

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**Desarrollo de una Interfaz Hombre-Máquina aplicado a un Sistema
de Detección de Fallos en una Planta Intercambiadora de Calor**

Tesis para optar el Título de **Ingeniero Electrónico**, que presenta el bachiller:

Gustavo Armando Orozco Loayza

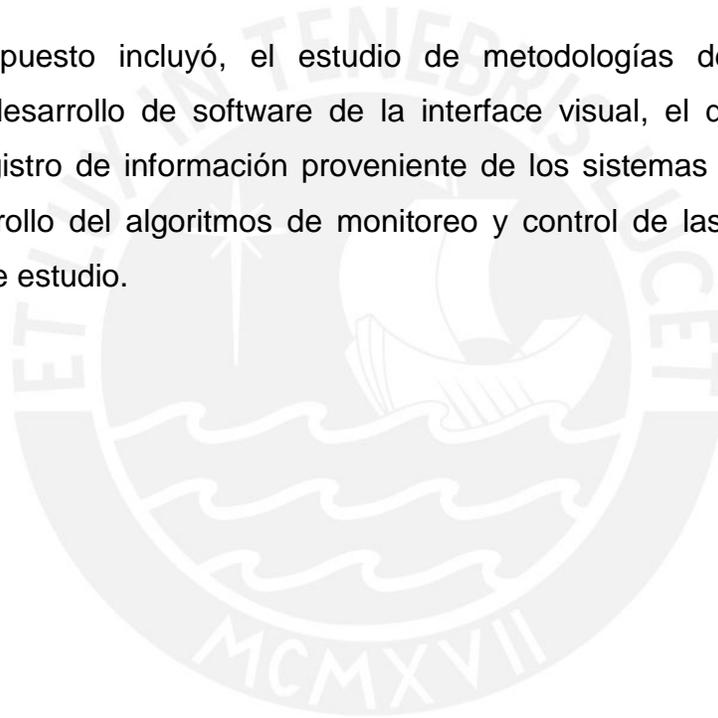
ASESOR: Dr. Juan Javier Sotomayor Moriano

Lima, Agosto de 2013

RESUMEN

La presente tesis tiene como objetivo el desarrollo de una interfaz hombre-máquina basada en una tarjeta de adquisición en una computadora. Esta interfaz, permitió visualizar y analizar la información proveniente de los sistemas de detección de fallos en los lazos de control de la planta estudiada. Asimismo, también permitió visualizar la evolución de las variables y actuar sobre estas cuando fue necesario.

El trabajo propuesto incluyó, el estudio de metodologías de desarrollo de interfaces, el desarrollo de software de la interface visual, el desarrollo de las librerías de registro de información proveniente de los sistemas de detección de fallos y desarrollo del algoritmos de monitoreo y control de las variables de la planta objeto de estudio.



ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: INTERFAZ HOMBRE-MÁQUINA PARA PLANTAS	3
1.1 INTRODUCCIÓN	3
1.2 DESCRIPCIÓN DE LA PLANTA INTERCAMBIADOR DE CALOR	4
1.3 SISTEMAS HOMBRE-MÁQUINA	6
1.4 CARACTERÍSTICAS DE LOS SISTEMAS HOMBRE-MÁQUINA.....	7
1.5 TIPOS DE CONTROL EN LOS SISTEMAS HOMBRE-MÁQUINA	8
1.6 TIPOS DE SISTEMAS HOMBRE-MÁQUINA.....	9
1.7 TIPOS DE HMI	10
1.8 CARACTERÍSTICAS BÁSICAS DEL HMI	11
1.9 INTERFAZ HOMBRE-MÁQUINA ORIENTADO AL CONTROL	11
1.10 FUNCIONES DE UN SOFTWARE HMI	12
1.11 TAREAS DE UN SOFTWARE DE SUPERVISIÓN Y CONTROL.....	13
1.12 ESTRUCTURA GENERAL DEL SOFTWARE HMI	13
CAPÍTULO 2 : DESCRIPCIÓN DE LOS MÉTODOS DE DESARROLLO DE INTERFAZ HOMBRE-MÁQUINA	16
2.1 FACTORES EN LA INTERACCIÓN HOMBRE-MÁQUINA	16
2.2 DISPOSITIVOS DE INTERACCIÓN.....	18
2.3 EL FACTOR HUMANO.....	20
2.4 INTERACCIÓN PERSONA-COMPUTADOR	21
2.5 INTERFAZ DE USUARIO	22
2.6 TIPOS DE INTERFACES DE USUARIO	23
2.7 REGLAS PARA EL DISEÑO DE INTERFACES DE USUARIO	25
2.8 PROCESO DE DISEÑO DE INTERFACES DE USUARIO.....	26
2.9 OBJETIVO DE LA TESIS.....	27
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ HOMBRE-MÁQUINA APLICADO A UNA PLANTA INTERCAMBIADOR DE CALOR	28
3.1 CONSIDERACIONES DE DISEÑO.....	28
3.2 DIAGRAMA DE BLOQUES DEL SISTEMA.....	36
3.3 DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS	38
3.4 ESCALAMIENTO DE SEÑALES	43
3.5 INTERFAZ DEL LAZO DE CONTROL	46
3.6 DISEÑO DEL ALMACENAMIENTO Y LECTURA DE DATOS.....	50
3.7 COMUNICACIÓN MATLAB - LABVIEW	56
3.8 COMUNICACIÓN LABVIEW - SIMULINK	57
3.9 DISEÑO FINAL DE LA INTERFAZ	59

CAPÍTULO 4: PRUEBAS Y ANÁLISIS DE RESULTADOS	61
4.1 INTERCAMBIADOR DE CALOR	61
4.2 ALMACENAMIENTO Y LECTURA DE DATOS DE LA VARIABLE DE PROCESO	61
4.3 ALMACENAMIENTO Y LECTURA DE DATOS DE LA TEMPERATURA EN EL CALENTADOR.....	64
4.4 DETECCIÓN DE FALLOS EN LA VÁLVULA	67
4.5 DETECCIÓN DE FALLOS EN LA BOMBA.....	69
4.6 INTERFAZ DE CONTROL	70
4.7 ANÁLISIS DE RESULTADOS	72
CONCLUSIONES	74
RECOMENDACIONES	75
BIBLIOGRAFÍA	76





INTRODUCCIÓN

Existen diversos métodos para la detección y diagnóstico de fallos en los procesos industriales. Un sistema de detección y diagnóstico de fallos es necesario en cualquier proceso automatizado para aumentar el grado de confiabilidad de los mismos.

El desarrollo de dicho sistema variará de acuerdo a la complejidad del mismo, a las habilidades y destrezas del programador, de los requerimientos del usuario y de las tareas que se realizarán y que estarán representadas por el software a utilizar.

Es necesario implementar algoritmos precisos para estar en la capacidad de detectar una falla en el momento que esta se encuentre en su fase de inicio y antes de que ocasione mayores daños a dispositivos libres de falla. Se busca también mediante este método evitar y detectar la ocurrencia de una posible falsa alarma en el proceso debido a perturbaciones internas o externas.

El trabajo propuesto tiene como finalidad el desarrollo de una interfaz hombre-máquina en base a señales proporcionadas por una tarjeta de adquisición de datos aplicado a un sistema de detección de fallos, el cual estará en la capacidad de detectar y dar la voz de alarma en caso ocurriera una falla en algunos de los componentes de la planta intercambiador de calor.

La interfaz gráfica constituirá la interacción entre la planta y el usuario y mediante la cual se podrá visualizar y analizar la información proveniente de los sistemas de detección de fallos en los lazos de control de la planta estudiada.

Así, el contar con aplicaciones de diagnóstico y detección que se ejecuten en tiempo real constituye un reto fundamental en el área industrial.

Los objetivos del trabajo incluyen la presentación y manipulación de la información obtenida de la planta, la visualización de las variables, así como

actuar sobre estas cuando sea necesario, el desarrollo de algoritmos de monitoreo y control sobre las variables de la planta objeto de estudio y finalmente realizar pruebas con la interfaz en la planta intercambiador de calor.

En el primer capítulo se describen sistemas de interacción hombre máquina, así como las características de la interfaz de estos sistemas orientados al control. Una descripción del diseño y desarrollo de interfaces hombre máquina es expuesto en el capítulo 2.

En el capítulo 3 se lleva a cabo el diseño de la interfaz hombre máquina aplicado a un sistema de detección de fallos en una planta intercambiadora de calor, basado en las librerías y herramientas que ofrece el entorno de programación LabVIEW.

En el capítulo 4 se llevan a cabo las pruebas y resultados que permiten observar el funcionamiento de la interfaz desarrollada.

Finalmente se realizan las conclusiones acerca de los trabajos realizados.

CAPÍTULO 1

INTERFAZ HOMBRE-MÁQUINA PARA PLANTAS

1.1 Introducción

En la actualidad, se han desarrollado interfaces entre el operador y los sistemas de control, basadas en dispositivos de computo, las cuales permiten acceder a la información acerca del funcionamiento de los equipos, la evolución de las variables, el estado de los procesos, etc.

En particular, para este fin se emplean interfaces basadas en computadoras, las cuales permiten acceder a la información que disponen los sistemas de detección de fallos, esta información por lo general está disponible en tiempo real y de manera visual, a fin de que el operador tome las decisiones más adecuadas.

La detección de fallas es de vital importancia en los procesos industriales, en estos procesos el funcionamiento adecuado de los equipos es un requerimiento imprescindible. Dado que estos procesos dependen del funcionamiento adecuado de los equipos que lo componen, el ser humano ha desarrollado gran preocupación por las condiciones en la que estos equipos se desenvuelven y operan, así como por la posible ocurrencia de una falla en el proceso y su respectiva localización, sea dentro o fuera de la planta.

A lo largo de la evolución, el hombre ha ido desarrollando diversos métodos para la detección de una posible falla en su entorno de trabajo, estos métodos se clasifican en los siguientes tres grupos:

1. Sentidos biológicos:

Se basan en la observación del proceso. Se tienen en cuenta aspectos como el cambio de forma, color, o la aparición de algún sonido extraño. También se pueden detectar fallas por medio del tacto, mediante la

sensibilidad a vibraciones o cambios de temperatura bruscos. Otro método se da por medio del sentido del olfato, pudiéndose percibir el olor característico del humo debido a fugas o sobrecalentamiento.

2. Sensores de medición:

Instrumentos que brindan información más exacta de las variables físicas involucradas en los procesos. Sin embargo demostraron ser propensos a malfuncionamiento, pues debido a cambios en los mismos procesos o en las condiciones ambientales se incrementaban los casos de falsas alarmas debido a una mala lectura del sensor. El potencial de este tipo de fallas aumentó críticamente cuando los sensores fueron aplicados en el control automático de las maquinarias, pues podían ocurrir daños más directos y devastadores.

3. La computadora y su aplicación en tiempo real:

Con la llegada de la computadora se produjo un desarrollo sustancial en los métodos de detección de fallas. Debido a su “inteligencia” y rapidez se puede contar con la integración de la información procedente de diversas fuentes, pudiéndose localizar el componente fallido en un sistema complejo incluyendo, si fuera el caso, el sensor causante de una falsa alarma. [1]

1.2 Descripción de la Planta Intercambiador de Calor

La planta tiene como objetivo principal controlar la temperatura de agua almacenada en un tanque de drenaje mediante un sistema intercambiador de calor.

En esta planta se controla el flujo de agua caliente que pasa por una tubería. Una bomba es la encargada de generar el movimiento del fluido. La bomba proporciona el flujo de agua de un tanque de agua caliente, cuya temperatura es regulada a través de un sistema de resistencias eléctricas. La temperatura transferida es controlada por la válvula proporcional que regula el ingreso de agua caliente al intercambiador.

Una segunda bomba proporciona un flujo de agua a temperatura ambiente de un tanque de suministro. La transferencia de calor producida por el paso de agua caliente en el intercambiador resultará en el incremento de la temperatura del flujo de agua a temperatura ambiente (la cual se desea controlar) y posteriormente se depositará en un tanque de drenaje.

En la figura 1.1 se muestra la planta intercambiador de calor instalada en el laboratorio de control y automatización, su respectivo diagrama de tuberías e instrumentación (P&ID) se muestra en la figura 1.2.

La interfaz Hombre-Máquina a desarrollar para la planta descrita permitirá:

- Visualizar y analizar la información proveniente de las señales obtenidas a través de la tarjeta de adquisición.
- Visualizar la evolución de las variables y actuar sobre estas cuando sea necesario.
- Desarrollo de algoritmos de monitoreo y control de las variables de la planta.
- Indicar el estado en el que se encuentran tanto la bomba centrífuga, así como la válvula neumática proporcional.



Fig. 1.1: Planta intercambiador de calor ubicada en el laboratorio Automatización.

DIAGRAMA P&ID DE LA PLANTA

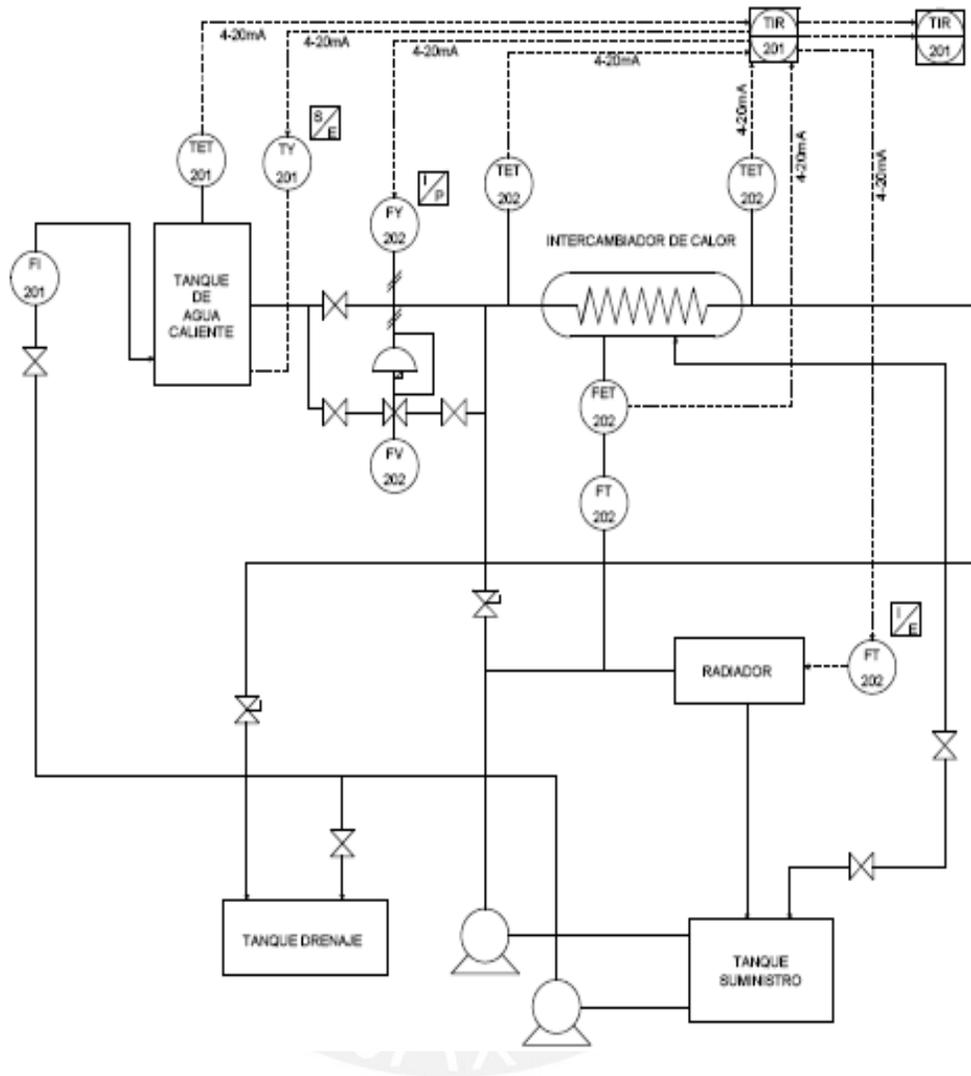


Fig. 1.2: Diagrama P&ID de la planta intercambiadora de calor

1.3 Sistemas Hombre-Máquina

Es la relación existente entre el hombre y la máquina cuando el primero se adapta a los requerimientos del segundo, de tal manera que la máquina dará información al hombre por medio de su aparato sensorial, el cual responde de distintas maneras si se altera el estado de la máquina mediante sus diversos controles. De esta forma, la información pasará de la máquina al hombre y otra

vez de éste a la máquina en un circuito cerrado de información y control. [2] En la figura 1.3 se detalla este tipo de sistemas.

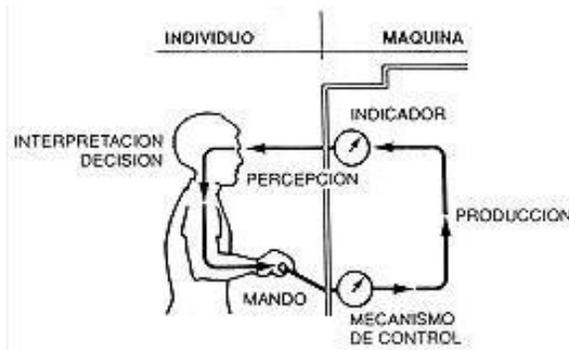


Fig. 1.3: Sistema Hombre-Máquina [19]

1.4 Características de los sistemas Hombre-Máquina

Para que un sistema cumpla su objetivo, ciertas funciones operacionales necesitan ser realizadas. La ejecución de cualquier función operacional a su vez acarrea una comunicación de cuatro funciones básicas:

A. Sentido:

Es la información que el sistema recibe del exterior. También cabe alguna información que se origine del interior del sistema, debida a una naturaleza de retroalimentación (feedback). El ser humano utiliza los sentidos como la vista y el tacto para percibir, la máquina tiene otros tipos de dispositivos, tales como sensores.

B. Almacenaje de información:

Para los seres humanos esto es simplemente un sinónimo de memoria de lo aprendido. En una máquina esto se puede guardar en diferentes componentes físicos y generalmente en forma codificada.

C. Procesamiento de la información y decisión:

Abarca varios tipos de operaciones realizadas con la información recibida y almacenada. Cuando los componentes de una máquina mecanizada o automatizada están implicados en el proceso informativo, dicho proceso tiene que estar programado para conseguir

que la máquina responda de una manera predeterminada a cada posible acceso de información.

D. Funciones de Acción:

Acciones u operaciones que resultan de las decisiones tomadas. Se realiza algún tipo de acción de control físico o de proceso, como la activación de ciertos mecanismos de control o el manejo, movimiento, modificación o alteración de materiales u objetos. Por ejemplo, variar el nivel de apertura de una válvula neumática proporcional en base a información recibida del proceso. [2]

1.5 Tipos de control en los sistemas Hombre-Máquina

Los sistemas de control de las máquinas automáticas pueden ser de lazo abierto o cerrado.

- Sistemas de lazo abierto:

Reciben las instrucciones y las ejecutan al pie de la letra, deteniendo su acción cuando los sensores reciben la información de que se ha cumplido con lo programado. En estos sistemas la variable controlada no es retroalimentada. [2] Este tipo de sistema se muestra en la figura 1.4.

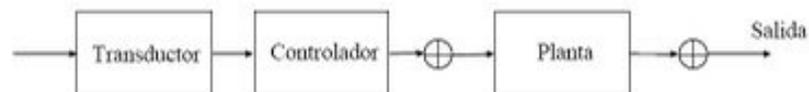


Fig. 1.4: Ejemplo de Sistema de Lazo Abierto

- Sistemas de lazo cerrado:

Son aquellos que tienen retroalimentación, esto quiere decir que cuando los sensores reciben información diferente a la que deben estar recibiendo tratan de corregir la operación del componente sobre el cual tienen acción. [2] Su principio de funcionamiento consiste en medir la variable controlada, convertirla en señal y retroalimentarla para compararla con la señal de entrada de referencia. La diferencia entre ésta señal y la señal

retroalimentada constituye la señal de error, la cual es empleada por el controlador para calcular la variación a aplicarse en la variable manipulada. [3] En la figura 1.5 se observa este tipo de sistema.

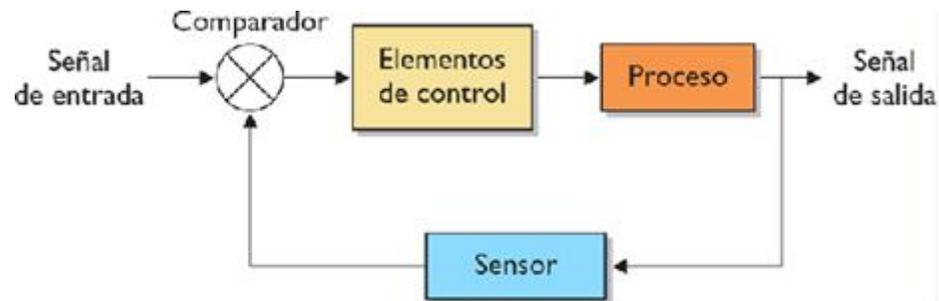


Fig. 1.5: Ejemplo de Sistema de Lazo Cerrado [2]

1.6 Tipos de sistemas Hombre-Máquina

Los sistemas pueden ser clasificados por los grados de control del hombre sobre la máquina.

1. Sistemas Manuales:

Constan de herramientas manuales y otras ayudas que se suman al operador humano que controla la operación. El operador transmite a sus herramientas y recibe de las mismas una gran cantidad de información, y en base a estas actúa a su propia velocidad.

2. Sistemas Mecánicos:

También llamados sistemas semiautomáticos, se componen de partes físicas con un buen nivel de integración. Disponen de herramientas mecánicas diseñadas para realizar sus funciones. La energía es proporcionada por la máquina y la labor por el operador. En este caso el operario recibe la información sobre el sistema por medio de visualizadores y realiza un proceso de información visual que sirve para tomar decisiones, las cuales se llevan a cabo empleando aparatos de control.

3. Sistemas Automáticos:

Realiza todas las funciones operacionales, incluyendo el sentido, el proceso de elaboración de la información y la toma de decisiones y de acciones. Se necesita de un sistema completamente programado para poder tomar medidas en caso de contingencias. La mayoría de estos sistemas son del tipo de lazo cerrado. Con estos sistemas las funciones humanas quedan reducidas a la de programación y mantenimiento. [2]

En base a estos sistemas de desarrollan distintos tipos de Interfaces Hombre-Máquina.

1.7 Tipos de HMI

Se describen los tipos de interfaz de usuario más comunes:

- Interfaces gráficas de usuario (GUI):
Aceptan la entrada a través de un dispositivo (mouse, teclado, etc.), proporcionando una salida gráfica en la pantalla del ordenador.
- Interfaces de usuario web:
Aceptan una entrada y proporcionan una salida a través de las páginas web transmitidas a través de internet. Los resultados serán vistos por el usuario mediante un navegador web.
- Pantallas táctiles:
Dispositivos que aceptan una entrada a través del tacto de los dedos o un lápiz táctil. Su uso más comercial se da en el campo de las telecomunicaciones.
- Interfaces de línea de comando:
El usuario proporciona la entrada al escribir desde un teclado una cadena de comando. A esta entrada el sistema proporciona una salida de impresión de texto en la pantalla del ordenador.

- Interfaces de voz del usuario:
Acepta la entrada y proporciona una salida mediante la generación de mensajes de voz. La entrada se realiza mediante la pulsación de teclas o botones, o respondiendo verbalmente a la interfaz. [4]

1.8 Características básicas del HMI

Los aspectos que se realizan en la comunicación hombre máquina son:

- A. Indicación del estado del proceso:
Para ello se pueden utilizar equipos convencionales, registradores, diodos emisores de luz, displays, etc. Para la selección en pantalla se pueden emplear: mouse, teclado, pantalla táctil, etc.
- B. Tratamiento e indicación de alarmas:
Se busca informar al operador de una situación anormal. Las alarmas pueden representarse:
 - En la propia pantalla, mediante símbolos que aparecen intermitentemente.
 - Mediante indicación sonora.
 - Mediante la impresión de mensajes de alarma.
- C. Ejecución de acciones de mando:
Se realizan por técnicas convencionales, es decir, a través de pulsadores, interruptores, potenciómetros, etc. Estos dispositivos de control se encuentran plasmados en la pantalla del ordenador y su activación representa una acción correctiva hacia el proceso real mediante una señal enviada a un elemento final de control. [4]

1.9 Interfaz Hombre-Máquina orientado al control

La sigla HMI es la abreviación en inglés de Interfaz Hombre-Máquina. HMI es una interfaz que permite la interacción entre un humano y una máquina, las cuales varían ampliamente, desde paneles de control para plantas nucleares

hasta botones de entrada en un celular. A los sistemas HMI en una computadora se les conoce como software HMI o de monitoreo y control de supervisión.

Las señales de los procesos son conducidos al HMI, tal como se muestra en la figura 1.6, por medio de dispositivos como tarjetas de entrada/salida en la computadora, PLC's, etc. [5]

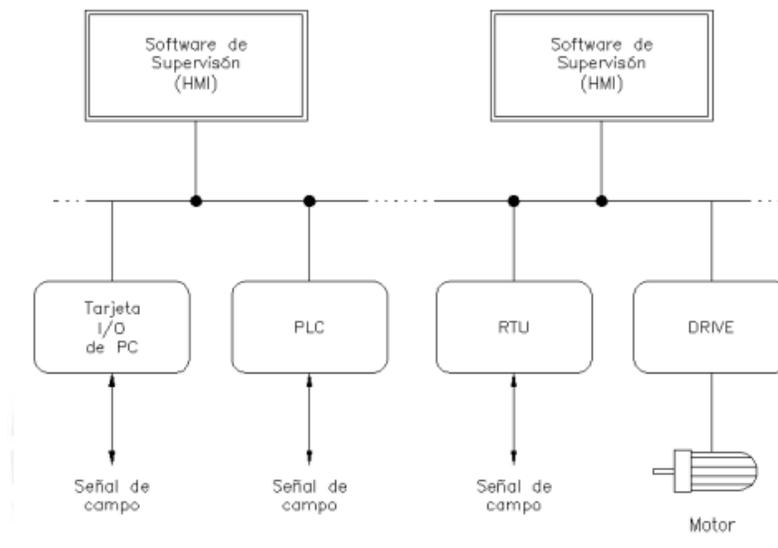


Fig. 1.6: Elementos principales de un Sistema HMI [5]

Una adecuada interfaz hombre-máquina persigue los siguientes objetivos:

- Captar la situación en forma rápida.
- Crear condiciones para la toma de decisiones correctas.
- Que los equipos se utilicen en forma rápida, óptima y segura.
- Garantiza la confiabilidad al máximo.
- Cambiar con facilidad los niveles de actividades del operador.

1.10 Funciones de un Software HMI

- Monitoreo: habilidad de obtener y mostrar datos en tiempo real. Estos pueden ser mostrados como número, texto o gráficos.

- Supervisión: permite ajustar las condiciones de trabajo directamente desde la computadora.
- Alarma: capacidad de reconocer eventos excepcionales dentro del proceso y a su vez reportarlos. Las alarmas reportadas están basadas en límites de control pre-establecidos.
- Control: capacidad de aplicar algoritmos que ajusten los valores del proceso y así mantenerlos dentro de ciertos límites. Esta función va más allá del control de supervisión, removiendo la necesidad de la interacción humana.
- Históricos: capacidad de muestrear y almacenar en archivos, datos del proceso a una determinada frecuencia. Este almacenamiento representa una herramienta poderosa a la hora de corregir y optimizar procesos. [9]

1.11 Tareas de un Software de Supervisión y Control

- Permitir una comunicación con dispositivos de campo.
- Almacenar la evolución de las variables del proceso en archivos almacenados en la computadora.
- Visualizar las variables mediante pantallas con objetos animados.
- Permitir que el operador pueda enviar señales al proceso mediante botones, controles ON/OFF, o con ajustes con el mouse o el teclado.
- Supervisar niveles de alarmas y alertar o actuar en caso de que las variables excedan límites de supervisión preestablecidos.

Para facilitar las tareas de diseño se incluyen en el software HMI gran cantidad de librería de objetos para representar dispositivos de uso en la industria, como: motores, tanques, indicadores, interruptores, etc. [5]

1.12 Estructura general del Software HMI

La estructura de un software HMI está compuesta por un conjunto de programas y archivos el cual se muestra en la figura 1.7.

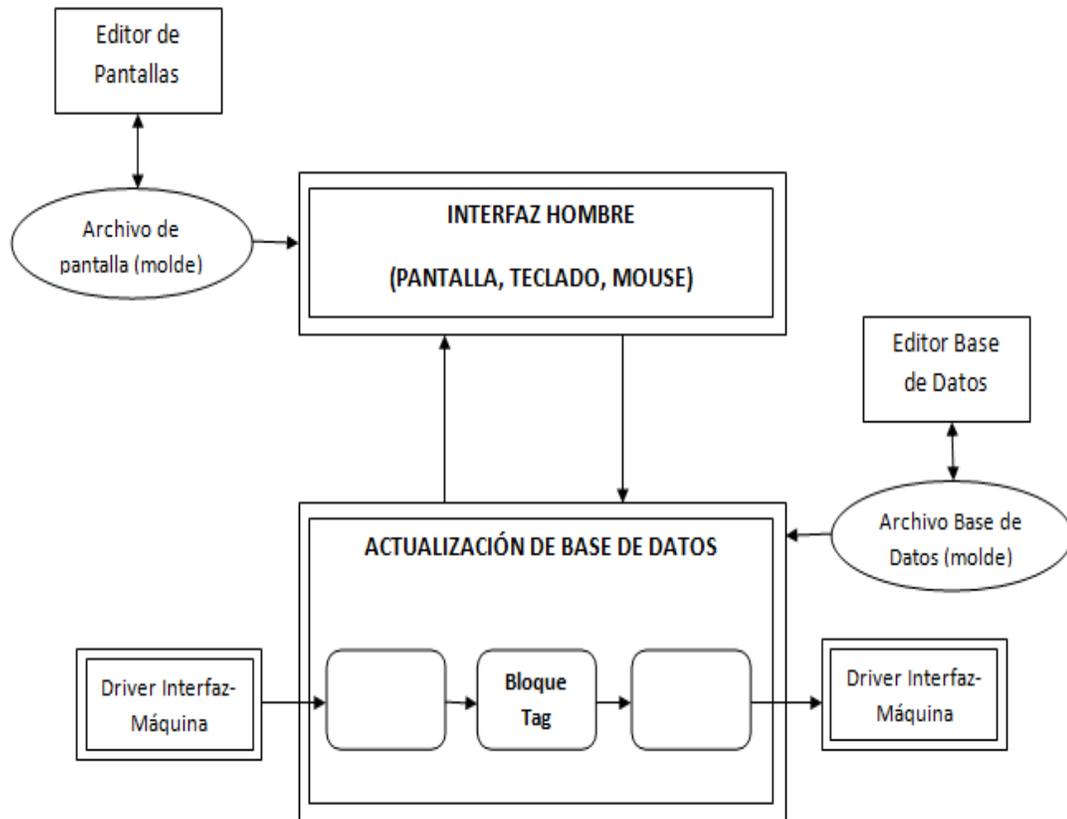


Fig. 1.7: Estructura general del Software HMI

Se muestra cómo funcionan algunos de los programas y archivos más importantes. Los rectángulos representan programas y las elipses representan archivos. Los programas que están con recuadro simple representan programas de diseño y lo que tienen doble recuadro representan programas que permiten el funcionamiento total del HMI.

Con los programas de diseño como el “Editor de Pantallas” se crean moldes de pantalla para la visualización de datos del proceso. Estos moldes serán guardados en archivos (“Archivos de Pantalla”) que almacenan la forma como serán visualizados los datos. [5]

La descripción de los demás bloques se da a continuación:

- **Interfaz Hombre:**
Es el programa que se encarga de refrescar las variables de la base de datos en pantalla y actualizarla. Constituye la interfaz entre la base de datos y el usuario.
- **Base de datos:**
Lugar en la memoria de la computadora donde se almacenan los datos requeridos del proceso.
- **Driver:**
Realiza la conexión entre los bloques de la base de datos y las señales del proceso. Manejan los protocolos de información entre el HMI y los distintos dispositivos de campo. Constituyen la interfaz hacia la máquina.
- **Bloques (tags):**
Se puede recibir información de los drivers o enviar información hacia los drivers. Será función principal de estos bloques comparar los valores adquiridos con valores umbrales de alarma.

Las funciones principales de los bloques son:

- Recibir datos de otros bloques o al driver.
- Enviar datos a otros bloques o al driver.
- Establecer enlaces a la pantalla.
- Realizar cálculos de acuerdo a las instrucciones del bloque.
- Comparar los valores con umbrales de alarmas.
- Escalar los datos del driver a unidades de ingeniería. [5]

CAPÍTULO 2

DESCRIPCIÓN DE LOS MÉTODOS DE DESARROLLO DE INTERFAZ HOMBRE-MÁQUINA

2.1 Factores en la interacción hombre-máquina

Hay problemas que necesitan ser planteados a la hora de diseñar un sistema de interacción hombre-máquina, tales como la salud de la persona, la eficiencia final del producto, el punto de vista social, etc.

Existen por tanto factores a considerar al momento de diseñar estos tipos de sistemas:

- Factores ergonómicos:

Es el estudio de las características físicas donde se produce la interacción. El objetivo es maximizar la seguridad, eficiencia y fiabilidad para simplificar las tareas e incrementar la sensación de confort.

Se consideran los siguientes aspectos:

- Organización de los controles y pantallas: permite una acción rápida del usuario.
- Información más importante situada a la altura de los ojos.
- Entorno físico de la interacción.
- Aspectos de salud, tales como posición física, tiempo de permanencia en el ordenador, temperatura, nivel sonoro, etc.
- Uso del color, deben usarse colores distinguibles.
- Legibilidad en pantalla.

La organización internacional de normalización (ISO) cuenta con grupos de trabajos para la elaboración de normas sobre ergonomía. Entre los cuales destacamos el WG5, que trata del diseño centrado en el hombre y los sistemas interactivos y el TC159/SC5, especializado en la ergonomía de la

interacción hombre-máquina. También podemos encontrar normativas sobre el diseño en la norma ISO 9241. [6]

- Factores psicológicos de los usuarios:

Hacen referencia a los procesos cognitivos de cada persona y al nivel de experiencia en el uso de sistemas informáticos en general o de uno en concreto. En los sistemas de interacción hombre-máquina es importante atender aspectos como la percepción, la atención, la memoria, el pensamiento y la resolución de problemas por parte de los operarios.

- Factores de diseño del sistema:

Es la actividad encaminada a conseguir la producción en serie de objetos útiles y visualmente agradables.

Se han diseñado lenguajes de programación de alto nivel, sistemas de gestión de interfaces de usuarios, entornos de diseño de estas interfaces, entre otras herramientas para esta finalidad.

El diseño de sistemas interactivos implica realizar un diseño pensando en el usuario, es decir, se implica al usuario tanto como sea posible, en algunos casos incluyéndolo en el equipo de diseño. En algunos sistemas es recomendable realizar prototipos para que el usuario pueda ir evaluando el diseño.

- Factores sociales y organizacionales:

Se refiere a los procesos de comunicación interactivos que se dan entre personas a través de máquinas. Su función es informar a los diseñadores acerca de la estructura social-organizacional de la institución y de cómo va a influir en sus miembros la introducción de la nueva tecnología. Al hablar de estructura social-organizacional abarcamos temas como la política de trabajo de la empresa, la manera de organizar las tareas y los roles de las personas, como influyen los miembros en el modo de utilizar los sistemas, el presupuesto, el equipamiento, etc.

Resumiendo, todos estos factores repercuten en el aumento o disminución de la productividad, de los costes, de los errores, del tiempo de producción y del surgimiento de ideas creativas e innovadoras para nuevos productos o acoplamientos que pueden mejorar el desempeño final del producto. Es por esto que hoy en día las empresas se preocupan de que sus empleados cubran sus necesidades y se encuentren a gusto al manejar los sistemas de interacción. [6]

2.2 Dispositivos de interacción

Podemos distinguir tres partes en el ordenador: los dispositivos de entrada de información, los dispositivos de salida y los que contienen a la unidad de procesamiento de información, sin el cual la interacción no sería posible.

Los que se presentan al usuario en el proceso de interacción son:

A) Dispositivos de entrada:

En este campo distinguimos dos tipos de dispositivos:

- Los de introducción de texto:

El usuario tiene que aprender comandos para interactuar con el sistema centrando por momentos su atención en el teclado y por otro en la pantalla.

Encontramos dispositivos en los que el usuario tiene un contacto físico con la pantalla: pantalla táctil y lápiz óptico.

- Los de indicación y selección:

Permiten la manipulación directa de los objetos que aparecen en pantalla, con la ventaja de que el usuario no tiene que aprender comando y que centra toda su atención directamente en la pantalla. Dentro de este campo encontramos dispositivos como: el ratón, el joystick, la tableta gráfica y el touchpad. [6]

Los más utilizados son:

- El teclado:

Es un dispositivo de entrada discreta, pues es sensible a una posición concreta. Es el más indicado para la introducción de texto en cantidades no

excesivamente grandes. Para ser manejable debe tener poco grosor y estar inclinado entre 10 y 25 grados.

- Pantallas táctiles:

Su uso no precisa aprendizaje y es altamente intuitivo e interactivo. Su principal desventaja radica en la comodidad física de su utilización, además de que produce manchas en la pantalla y precisa separar las manos del teclado. Orientados a contextos con usuarios que las van a usar durante poco tiempo. Ejemplo: cajeros automáticos en bancos, puntos de información, etc.

- Ratón:

Indica partes de la pantalla de forma indirecta. Su principal ventaja es que no se produce tanta fatiga en el brazo, pero se precisa una coordinación entre el ojo y la mano. Permite mover el puntero por toda la pantalla, así como seleccionar una parte del mismo.

- Reconocimiento del habla:

Es el más natural para las personas y el más fácil de usar, ya que no precisa ninguna formación o instrucción previa. Reconoce el habla humana y la reproduce para facilitar el trabajo del hombre con la máquina. Resultan de utilidad cuando el usuario tiene las manos ocupadas, o si presenta alguna discapacidad física que le impida manejar el ordenador de la forma convencional (teclado y ratón).

B) Dispositivos de salida:

- Visuales:

Hacen referencia a que la lectura o visualización de la información sea desarrollada de forma cómoda. Este aspecto incluye:

- Aspectos físicos:

Tales como el brillo, el contraste entre los caracteres y el fondo, la combinación de colores, el tamaño de los caracteres, los íconos y el orden de los ítems en el menú.

- Aspectos técnicos:

Referentes a la resolución y el parpadeo producido por un lento refresco de pantalla.

Un mal diseño que resulte en la baja calidad de alguno de estos aspectos harán las tareas algo fatigantes para el ojo, conllevando posiblemente a una interpretación errónea de lo que se percibe.

- Sonoros:

Puede resultar de gran utilidad si los dispositivos representan determinados mensajes y el usuario conoce el significado de cada uno. Se incluyen gamas de sonido en que el usuario elige según la finalidad para los que se requiera: abrir un documento, eliminarlo, enviar un mensaje, avisar de un error, etc. Se le da utilidad en aplicaciones en las que el usuario tiene la vista fijada en algún punto en concreto fuera o dentro de la pantalla, desatendiendo por momentos el espacio donde puedan aparecer mensajes. [6]

2.3 El factor humano

Se abarca principalmente el modelo mental que tiene el usuario del sistema de interacción en comparación al modelo que el diseñador ha creado para ese sistema.

Al diseñar interfaces de usuario deben tenerse en cuenta las habilidades cognitivas y de percepción de las personas y adaptar el programa a ellas. Es por eso que al momento en que el diseñador se plantea la creación del sistema debe mantener en prioridad desarrollarlo al servicio del usuario. Para cubrir esta prioridad se debe pensar en cómo lo va a utilizar el usuario al que va dirigido y si con el diseño se logra plasmar en el usuario el modelo tal y como el diseñador lo ha pensado y desarrollado. [7]

Para lograr estos fines IBM en 1992 ilustra el trabajo que debe hacer el diseñador comparándolo con un iceberg. Se explica que de todo el bloque de hielo el diseñador debe dedicar sólo un 10% a lo que quedará visible, es decir, a las representaciones visuales y a la estética de la interfaz. Del 90% restante

deberá dedicar un 30% a las formas de comunicación entre el usuario y el sistema, y el 60% final a lo que menos verá el usuario, es decir, los propios objetos y sus propiedades y comportamientos.

De esta forma siguiendo el esquema del “look-and-feel iceberg” propuesto por IBM, tal como se muestra en la figura 2.1, se conseguirá plasmar de una forma real el modelo del sistema en la mente del usuario. Por consiguiente se evita que se cree un modelo falso dificultando el entendimiento del sistema. [6]



Fig. 2.1: Esquema “look-and-feel iceberg” propuesto por IBM (1992) [6]

2.4 Interacción persona-computador

Estudia el intercambio de información mediante software entre las personas y las computadoras.

Los componentes de este sistema de interacción son:

1. El usuario:
Persona con capacidad limitada de procesar información a través del razonamiento y de habilidades adquiridas.
2. El computador:
Máquina que cuenta con diversos tipos de dispositivos de entrada y salida que lo mantienen en contacto con el mundo físico.

Para el desarrollo de una **interfaz** que se encargue de este tipo de interacción, tal como se muestra en la figura 2.2, se tomará como eje de diseño fijar quien o quienes serán los usuarios y las tareas que se realizarán. [8]

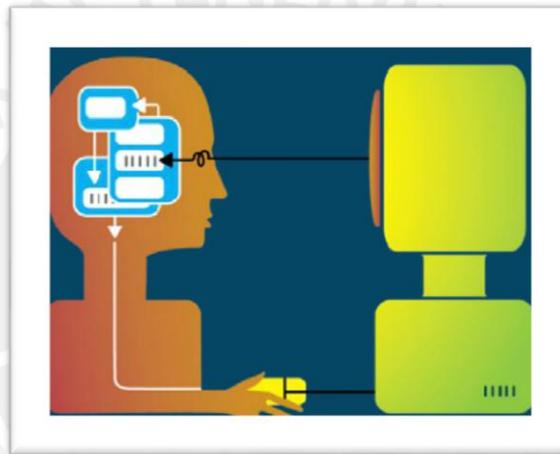


Fig. 2.2: Modelo Mental de interacción Persona-Computador [8]

2.5 Interfaz de usuario

En términos generales, una **interfaz** es el área o superficie a largo de la cual dos cosas de naturaleza distinta convergen. Basándonos desde esta perspectiva denominamos interfaz a cualquier medio que permita la interconexión de dos procesos diferenciados con un único propósito. [9]

Se entiende la interfaz como el conjunto de comandos y métodos que permiten la intercomunicación del programa con cualquier otro programa o elemento

interno o externo, los cuales representan a los periféricos que estarán controlados por las interfaces. [10]

Una vez definida la idea de lo que representa la interfaz, podemos hablar ahora de **interfaz de usuario**.

En software, una interfaz de usuario es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. Está constituida por una serie de dispositivos, tanto físicos como lógicos, que permitirán al hombre interactuar de una manera precisa con un sistema.

Para que un sistema interactivo, como las interfaces de usuario, cumpla sus objetivos tiene que ser usable y accesible a la mayor parte de la población humana. Se define usabilidad como la medida en que un producto se puede usar por determinados usuarios para conseguir unos objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso dado. Habitualmente cuando se prima la estética sobre la usabilidad el resultado es un objeto mal diseñado. [11]

Interfaces de usuario de un programa:

Conjunto de elementos hardware y software de un ordenador que presentan información al usuario y le permiten interactuar con la información y con el ordenador. Incluye el hardware que incluye el sistema: teclado, dispositivo apuntador (ratón), monitor, etc. [9]

Los componentes software son los elementos que el usuario ve, oye, a los que apunta o toca en la pantalla para interactuar con el ordenador, así como la información con la que trabaja.

2.6 Tipos de Interfaces de usuario

- Interfaces de línea de mandatos:
 - por sus siglas en inglés CUIs (Command_line User Interface).
Corresponde al estilo más antiguo de interacción hombre-máquina. Se

usaba un lenguaje formal con vocabulario y sintaxis propia. En este tipo de interfaces el usuario no recibía mucha información por parte del sistema. Es característico de los sistemas DOS.

Desventajas:

- Carga de memoria del usuario.
- Inflexible en la sintaxis.
- No adecuado para usuarios noveles.

Dentro de sus ventajas encontramos:

- Adecuado para usuarios experimentados.
-
- Interfaces de menús:
Se plasmaban lista de opciones que se mostraban en pantalla o en una ventana para que los usuarios elijan la opción que deseaban.
Permitían navegar dentro de un sistema y seleccionar elementos de una lista que representan propiedades o acciones que los usuarios precisen realizar.
 - Interfaces gráficas:
O por sus siglas en inglés GUIs (Graphical User Interfaces). Es un método para facilitar la interacción del usuario con el ordenador a través de la utilización de un conjunto de imágenes y objetos pictóricos además de texto. Constituye el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema de control o informático. Las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora.
- Características de un GUI:
- Posee monitor gráfico de alta resolución.
 - Posee dispositivo apuntador.
 - Promueve la consistencia de la interfaz entre programas.

- Permite manipular en pantalla directamente objetos e información.
- Posee elementos de interfaz estándar como menús y diálogos.
- Posee muestra visual de información y objetos (íconos y ventanas).
- Proporciona respuesta visual a las acciones del usuario.
- Existen controles gráficos para la selección e introducción de la información. [10]

2.7 Reglas para el diseño de interfaces de usuario

Mediante el desarrollo de una interfaz de usuario se podrán integrar el conjunto de elementos hardware y software que presentan la información y que le permitirán al operario interactuar con el ordenador.

Principios a seguir en el desarrollo de interfaces de usuario:

1. Dar control al usuario:

- Permite una cómoda navegación dentro del producto y una fácil salida del mismo.
- Permitir al usuario interrumpir su tarea y continuarla más tarde.
- Utilizar mensajes y textos descriptivos.
- Permite al usuario manipular directamente los objetos de la interfaz.
- Hacer “transparente” la interfaz del usuario, es decir, este debe creer que trabaja directamente con los objetos.
- Permitir al usuario personalizar la interfaz.

El usuario debe sentir que tiene control del sistema.

2. Reducir la carga de memoria del usuario:

- Dar indicaciones visuales de donde está el usuario, que está haciendo y que puede hacer a continuación.
- Asociar acciones a los objetos (menú de barras, de herramientas).
- Presentar al usuario únicamente la información que necesita.
- Hacer una presentación visual clara.
- Proporcionar atajos de teclado.

- Presentar al usuario sólo la información que necesita.

3. Consistencia:

Un sistema es consistente si todos los mecanismos que se utilizan son siempre usados de la misma manera sea cual sea el momento en que se utilicen. Se recomienda para este propósito no hacer modificaciones si no es necesario hacerlas.

Se le da consistencia al sistema en los siguientes aspectos:

- En la realización de tareas: proporcionar indicaciones al usuario sobre el proceso que se está haciendo.
- En el resultado de las acciones: se busca una misma respuesta ante una misma acción.
- En la apariencia estética: íconos, fuentes, colores, distribución de pantalla.

Con la introducción de estos tres aspectos en el diseño de la interfaz de usuario se busca fomentar la libre exploración de la interfaz sin miedo a consecuencias negativas. [10]

2.8 Proceso de diseño de interfaces de usuario

Este proceso se divide en las siguientes cuatro fases:

- Reunir y analizar la información del usuario:
En este punto se definen: los usuarios a los cuales está destinado el sistema, las tareas que se van a realizar y como se van a realizar, que es lo que exigen los usuarios del producto y cuál será el entorno de desarrollo tanto de la interfaz como del usuario.
- Diseñar la interfaz de usuario:
Se definen los objetivos de usabilidad, los objetos que irán en la interfaz y sus respectivas acciones, la cantidad de íconos y formas visuales que

ayudarán al usuario a disminuir su carga de memoria y el menú de objetos y ventanas de los cuales dispondrá el producto final.

- Construir la interfaz de usuario:
Se construye un prototipo de la interfaz en base a un lenguaje de programación de alto nivel. Una vez construido y probado el prototipo se procede a la implementación del mismo.
- Validar la interfaz de usuario:
Se realizan las pruebas finales de usabilidad del sistema, mucho mejor si los usuarios finales participan de estas pruebas. [10]

2.9 Objetivo de la Tesis

La presente tesis tiene como objetivo el desarrollo de una interfaz hombre-máquina que permita visualizar la detección de fallos en los distintos dispositivos y componentes en una planta intercambiador de calor. Esta interfaz se desarrollará a través del entorno de programación LabVIEW, un sistema de programación gráfica para aplicaciones que involucran adquisición, control, análisis y presentación de datos.

Los objetivos específicos incluyen:

- Implementar mediante el uso de LabVIEW una interfaz hombre-máquina que interactúe con el sistema de detección de fallos.
- Desarrollar una interfaz gráfica para presentar y manipular la información y los instrumentos propios de la planta en tiempo real.
- Visualizar la evolución de las variables y actuar sobre estas cuando sea necesario.
- Desarrollo de algoritmos de monitoreo y control de las variables de la planta objeto de estudio.
- Realizar pruebas con la interfaz desarrollada en la planta intercambiador de calor.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ HOMBRE-MÁQUINA APLICADO A UNA PLANTA INTERCAMBIADOR DE CALOR

3.1 Consideraciones de diseño

Para el desarrollo del trabajo propuesto se usará un lenguaje de programación donde los usuarios puedan realizar sus aplicaciones de manera rápida y con una interfaz gráfica agradable.

Basándonos en este propósito se elige el entorno de programación ofrecido por LabVIEW para desarrollar la interfaz gráfica con la cual interactuará el usuario.

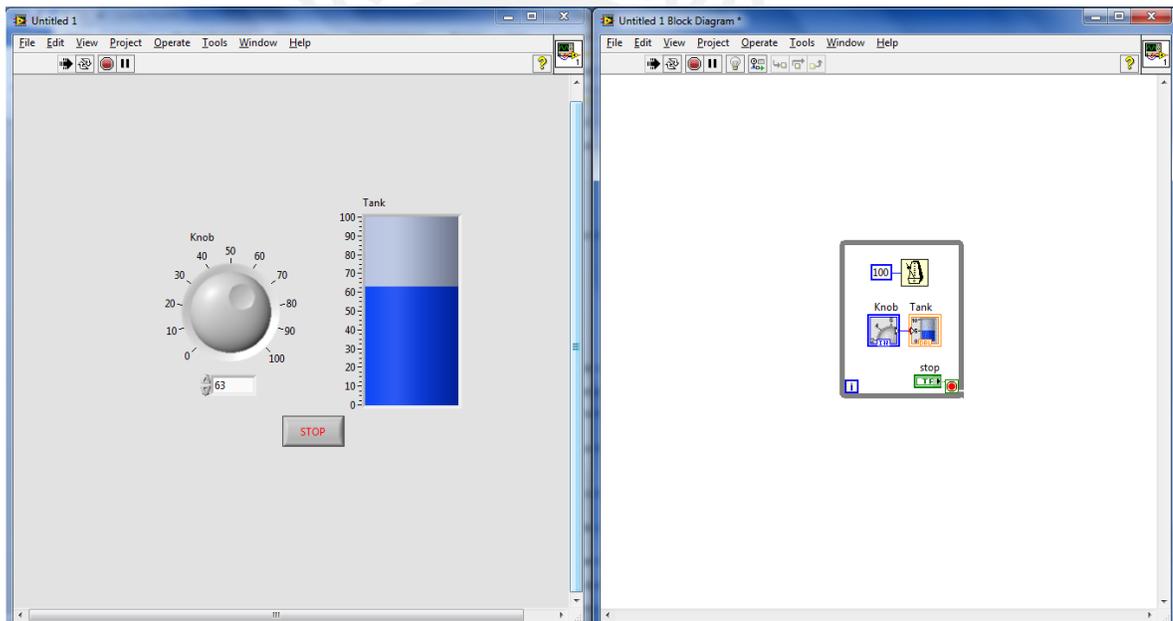
LabVIEW es el acrónimo de “Laboratory Virtual Instrument Engineering Workbench”. Es un sistema de programación gráfico para aplicaciones que involucran adquisición, control, análisis y presentación de datos.

Las ventajas que proporciona este entorno de programación son las siguientes:

- Reduce el tiempo de desarrollo de las aplicaciones, debido a que su programación es intuitiva y fácil de usar.
- El sistema cuenta con un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Se puede incorporar aplicaciones desarrolladas en otros lenguajes de programación. Sobre todo de C++.
- Permite realizar simulaciones en tiempo real y analizar los resultados antes de ser implementados a la realidad. [12]

Entorno LabVIEW

En sus inicios este programa estaba orientado a aplicaciones de control de instrumentos electrónicos usados en el desarrollo de sistemas de instrumentación, lo que se conoce como instrumentación virtual. Es debido a este motivo que los programas creados en LabVIEW se guardan en ficheros llamados VI (Virtual Instrument) y con la misma extensión. Partiendo de este concepto se le da nombre a sus dos ventanas principales. El “Panel Frontal”, donde estarán los botones, pantallas, etc., y el “Diagrama de Bloques”, donde estará representada la circuitería interna que le da lógica a los botones, pantallas, etc., plasmados en el panel frontal. [13]



Panel de Frontal

Diagrama de Bloques

Fig. 3.1: Instrumento Virtual creado en LabVIEW

De la figura 3.1 podemos concluir que el uso de lenguaje de programación G (gráfico) de LabVIEW permite su utilización en personas con conocimientos limitados de programación, pues como observamos los programas no se escriben, sino, se dibujan.

Consideraciones Hardware

Tarjeta de adquisición de datos

Se encarga de las etapas de adquisición y generación de señales a través de una tarjeta de adquisición de datos (tarjetas DAQ). Estas tarjetas necesitan de un ordenador para gobernarlas. Encontramos de dos tipos:

- Tarjetas DAQ internas: usan como interfaces más habituales PCI, PXI o PCI Express.
- Tarjetas DAQ externas: como USB o RS-232 e incluso inalámbricas. [13]

En la figura 3.2 identificamos las partes de un sistema DAQ:

- Señales de entrada y salida, tanto analógicas como digitales.
- Dispositivo de adquisición de datos (tarjeta DAQ).
- Software de aplicación (entorno LabVIEW). [14]



Fig. 3.2: Partes de un sistema DAQ [14]

La tarjeta DAQ presenta las siguientes etapas:

- Etapa multiplexora:
Se encarga de recibir las señales provenientes del proceso para después seleccionar un canal de entrada. Funciona como un dispositivo de conmutación con el que las entradas se muestran por turnos.
- Acondicionamiento de la señal:
Consta de dos etapas:

- **Amplificación:**
Se encarga de darle a la señal de entrada un nivel adecuado para su posterior procesamiento.
- **Filtrado:**
Se refiere al proceso de eliminación de cierta banda de frecuencias de una señal.

Conversión Analógica-Digital:

Tiene dos etapas:

- **Circuito de Muestreo y Retención:**
Captura la señal de entrada y la mantiene constante durante un ciclo de conversión analógica-digital.
- **Convertor A/D:**
Implica la conversión de las señales analógicas a palabras binarias manipuladas fácilmente por la computadora.

Conversión Digital-Analógica:

El convertor D/A recibe como entrada una palabra binaria. La salida es una señal analógica que representa la suma ponderada de los bits que no son cero representados en la palabra binaria de entrada. [15]

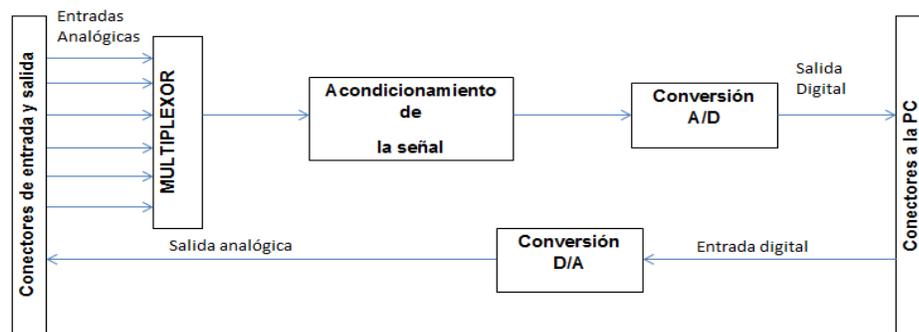


Fig. 3.3: Arquitectura de una tarjeta DAQ

En la figura 3.3 se muestra la arquitectura de una tarjeta de adquisición de datos.

El laboratorio de Control y Automatización tiene instalado en cada computadora una tarjeta DAQ multifunción PCI-6229 fabricada por National Instruments. Es con esta tarjeta que se desarrollará la adquisición y transmisión de señales.

Las características técnicas se detallan en la tabla 3.1.

Parte Analógica	Entrada	N° Canales	32 en modo común 16 en modo diferencial
		Resolución	16 bits
		Velocidad de muestreo	250 KS/s
		Rango de Voltaje de Entrada	(+/- 0.2 V, +/- 1 V, +/- 10 V, +/- 5 V)
	Salida	N° Canales	4
		Resolución	16 bits
		Velocidad de muestreo	833 KS/s
		Rango de salida	(+/- 10V)
Parte Digital	Entrada/Salida	N° Canales	48
		Niveles lógicos	TTL
		Contadores/Temporizadores	2

Tabla 3.1: Características técnicas de la tarjeta DAQ PCI-6229

En la figura 3.4 se muestra la tarjeta DAQ usada en el laboratorio.



Fig. 3.4: Tarjeta DAQ multifunción PCI-6229

Sensores de temperatura en la planta intercambiador de calor

La medición de temperatura constituye una de las mediciones más importantes realizadas en los procesos industriales.

Se utiliza una gran variedad de transductores, entre los más comunes encontramos: bandas bimetálicas, termopares, termistores, sensores de semiconductor, pirómetros de radiación y los detectores de temperatura resistivos (RTD).

En la planta intercambiador de calor se tienen acoplados sensores de temperatura del tipo RTD. En estos tipos de sensores tenemos las siguientes características:

- El cambio en la resistencia de un metal hace posible medir la temperatura a través del paso de corriente eléctrica.
- Se caracterizan por tener buena linealidad en el rango de operación.
- Los rangos de temperatura varían según el metal utilizado.
- Alta relación señal-ruido, lo que permite mejorar la precisión en la medida permitiendo distancia mayores entre el sensor y el equipo de medición. [13]

El sensor tipo RTD que se utiliza es del tipo Pt-100. Estos son sensores que trabajan midiendo la resistencia eléctrica a través del metal platino (Pt). Trabajan en rangos de temperatura entre -200°C y 600°C , en donde la resistencia del metal aumenta en función de la temperatura. La resistencia a 0°C es de 100 ohmios, es por esta razón que se le da el nombre de Pt-100. [3]

Los sensores de este tipo conectado con cables de mayor longitud pueden proporcionar lecturas imprecisas. Desde este punto de vista la resistencia del cable introduce dos tipos de errores, un error de desviación causada por la resistencia del cable en sí y un cambio en la resistencia del cable en relación a la variación de temperatura. Este error se supera mediante el uso de sensores de tres cables, donde el tercero mide la resistencia existente en el cable. Se usa el primer cable para alimentar al sensor a corriente constante, por el segundo

se mide la tensión y el cable número tres sirve como referencia y retorno de corriente. En este tipo de configuración la resistencia de los cables uno y dos no afecta la medida. [13]

El rango de salida del transductor Pt-100 se encuentra estandarizado por la industria en valores de corriente de 4 a 20 mA. En la figura 3.5 se muestra el sensor de temperatura instalado en la planta.



Fig. 3.5: Transmisor de temperatura Pt-100 ubicado en la planta

Actuadores en la planta intercambiador de calor

Válvula de control

La válvula de control es el elemento final de regulación utilizado comúnmente en toda malla de control. Se comportan como un orificio de restricción variable, su función consiste en controlar, de acuerdo con una señal, el caudal de un fluido de proceso. El actuador de la válvula convierte la señal de entrada en desplazamiento lineal, el cual es transmitido al cuerpo para variar la apertura o cierre y consecuentemente el caudal.

La planta tiene instalada un válvula tipo globo, en el cual el cierre se logra por medio del disco que corta el paso del fluido en un asiento que suele estar en paralelo con la circulación del mismo, realizando un control preciso. [3]

En la planta intercambiadora de calor se va a diagnosticar la falla en una válvula proporcional de accionamiento neumático de la compañía Fisher mostrada en la figura 3.6.



Fig. 3.6: Válvula neumática-proporcional instalada en la planta

Bomba centrífuga

Tienen como función principal el mover el volumen de un líquido entre dos niveles. En esencia son máquinas hidráulicas que transforman trabajo mecánico en trabajo hidráulico. Las bombas centrífugas son las de mayor uso en las plantas industriales debido a su alta eficiencia y al bajo costo de mantenimiento que se les brinda. [3]

En la planta intercambiador de calor se va a diagnosticar la falla en una bomba centrífuga Cpm 130 de la compañía Pedrollo mostrada en la figura 3.7, la cual trabaja con un motor eléctrico monofásico de 0.75 Hp y 3450 RPM.



Fig. 3.7: Bombas centrífugas instaladas en la planta

3.2 Diagrama de Bloques del Sistema

El desarrollo de la interfaz estará basado en la integración de todas las variables requeridas por el sistema de detección de fallos que cumplan con los requerimientos planteados.

Identificamos tres etapas para cubrir dichos requerimientos:

Interfaz de control

Identificamos cuatro bloques, los cuales se muestran en la figura 3.7. El primero se encargará de adquirir la señal del sensor de temperatura de la variable de proceso de la planta, en este caso de la temperatura del agua que va al tanque de drenaje. El bloque de escalamiento se encarga de encontrar la ecuación de la recta que relaciona la señal eléctrica obtenida del sensor de temperatura y su respectivo valor en grados centígrados. El procesamiento de datos hace referencia al algoritmo de control que se le va a aplicar a la señal escalada, en este caso se usará el algoritmo PID, también abarca etapas como la de almacenamiento y lectura de la evolución de las señales adquiridas. Finalmente tenemos el bloque de interacción hombre- máquina, etapa en que se le da control de la interfaz al usuario y consecuentemente interacción con la planta.

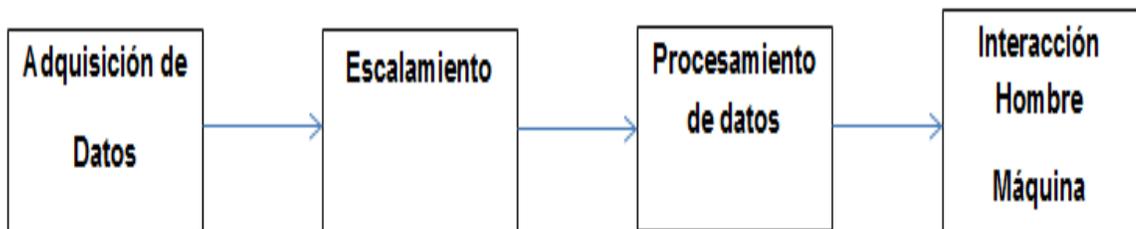


Fig. 3.8: Diagrama de Bloques del Control del Proceso

Sistema de detección de fallos para la válvula

En esta etapa identificamos cinco bloques, los cuales se muestran en la figura 3.9. El primero corresponde a la adquisición de datos, etapa en la cual se

trabaja con la señal de temperatura correspondiente al agua almacenada en el tanque de drenaje. Se procede a realizar el escalamiento pertinente a dicha señal. A continuación encontramos la etapa de procesamiento, en la cual se almacenan la variable de proceso (PV), variable de control (OP) y el set point (SP) del controlador PID implementado. El cuarto bloque utiliza los valores de estas variables para la ejecución del algoritmo, el cual será ejecutado en el entorno de MATLAB. Finalmente tenemos la etapa que se encarga de mostrar el resultado de la ejecución del algoritmo de detección en LabVIEW.

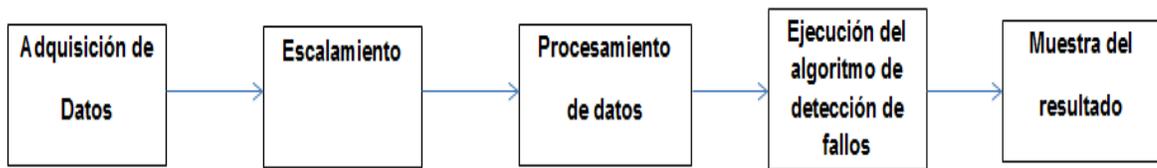


Fig. 3.9: Diagrama de Bloques del S. de Detección de Fallos para la Válvula
Sistema de detección de fallos para la bomba

Se identifican tres bloques, los cuales se muestran en la figura 3.10. El primero corresponde a la comunicación que se entabla entre los entornos de programación que ofrecen Simulink y LabVIEW. A continuación se procede a ejecutar el algoritmo de detección de fallos para la bomba centrífuga en Simulink para finalmente plasmar el resultado obtenido en la interfaz provista por LabVIEW.

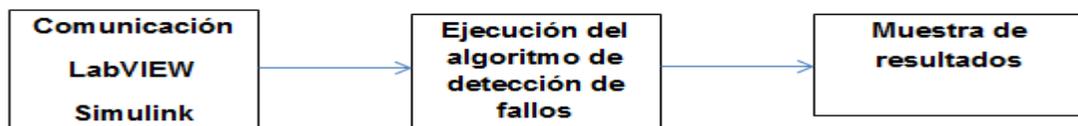


Fig. 3.10: Diagrama de Bloques del S. de Detección de Fallos para la Bomba Centrífuga

3.3 Diseño del sistema de Adquisición de Datos

Para posibilitar la adquisición de datos en LabVIEW se precisa del driver correspondiente para la fácil comunicación entre el software y el hardware. El driver que utiliza LabVIEW para este propósito recibe el nombre de NI-DAQmx, la cual proporciona una librería que facilita la comunicación con tarjetas de adquisición.

En la figura 3.11 se presenta la paleta asociada al NI-DAQmx.

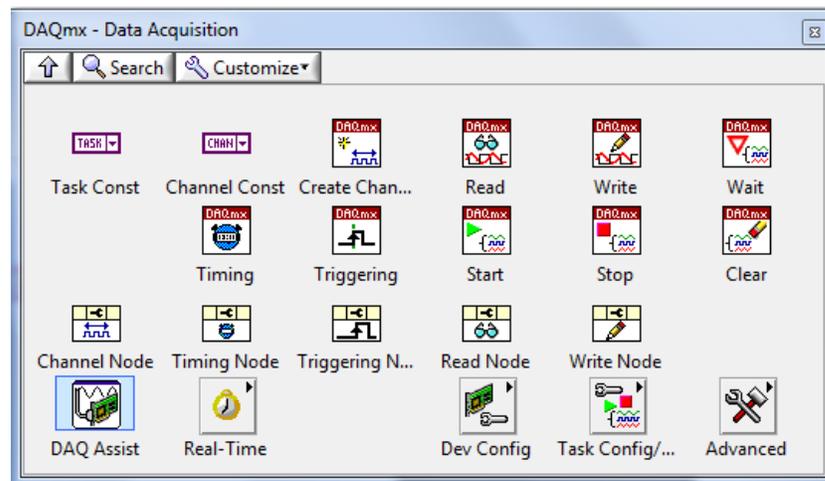


Fig. 3.11: Funciones para trabajar con tarjetas DAQ

Con esta librería o API (Application Programming Interface) podremos configurar la tarjeta de adquisición, leer los valores de las señales que llegan a los canales entrada, así como escribir valores en los canales de salida tanto analógica como digital. [13]

Para el proceso de adquisición se usarán los siguientes VI:

DAQmx Create Channel

Como observamos en la figura 3.12, la instancia a usar, sea de entrada o salida analógica o de entrada o salida digital, se selecciona mediante una lista desplegable que aparece debajo del ícono. Al crear una constante o control para seleccionar en que canal deseo recibir o enviar datos (terminal de conexión “physical channels”), debe aparecer una lista desplegable en la que se

muestran los que están disponibles. Para seleccionar el canal se sigue el formato “Dev1/ai0”, donde 1 es el número de dispositivo y ai0 es el número de canal, en este caso “analog input 0”. Si se quisiera leer más de una entrada analógica a la vez se sigue el formato “Dev0/ai0:1”, el cual posibilita leer los datos de las entradas analógicas 0 y 1 a la vez siempre y cuando se configure el “DAQmx Read” en la opción de canales múltiples. [16]

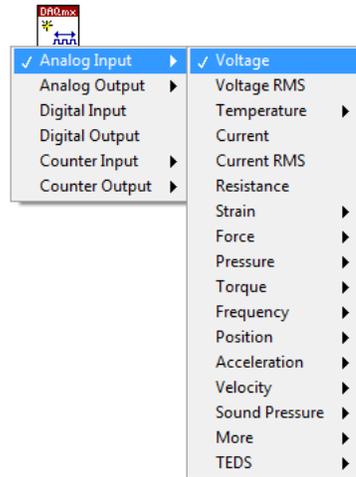


Fig. 3.12: Configuración para entrada analógica de tensión

En la figura 3.13 se muestra el DAQmx Create Channel, sobre el cual encontramos la opción “input terminal configuration”, el cual sirve para indicar el tipo de medida. Encontramos el modo diferencial, RSE o NSRE.

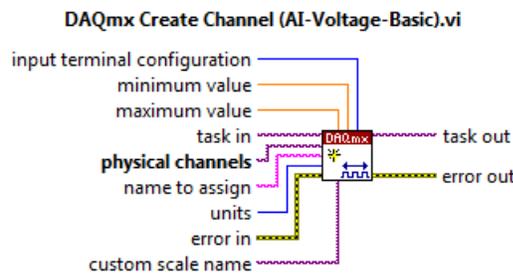


Fig. 3.13: DAQmx Create Channel

DAQmx Start Task

Se encarga, una vez creado el canal, de iniciar la adquisición de datos o de iniciar la generación de señales. Es decir, pone en el canal en estado de operación activa. En la figura 3.14 se muestra su representación en el diagrama de bloques.

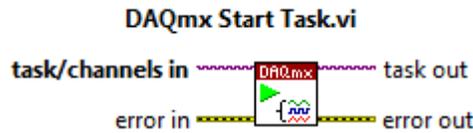


Fig. 3.14: DAQmx Start Task

DAQmx Read

En la figura 3.15 se muestra su representación en el diagrama de bloques. Su función es leer una o varias señales por canal, esto depende de la selección que se hace en una lista desplegable debajo del ícono, seleccionando “Single Channel” estaremos configurando el canal para la lectura de una sola señal, y a fin de tener lectura de múltiples señales se selecciona “Multiple Channels”. [13]

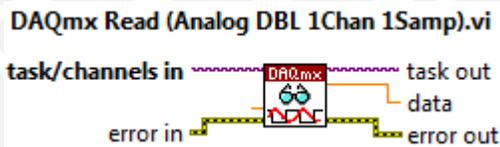


Fig. 3.15: DAQmx Read

Al seleccionar una instancia en concreto, tal como se muestra en la figura 3.16, se determina el formato de la lectura.

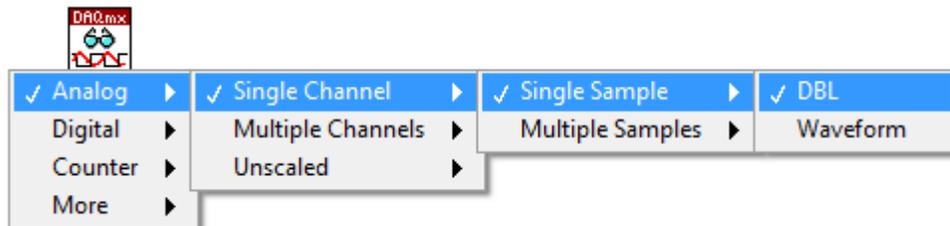


Fig. 3.16: Estableciendo el formato de la lectura en el canal

DAQmx Write

Este VI se encarga de generar señales en el canal especificado. En funcionamiento por defecto de este VI es esperar a generar todas las muestras antes de seguir con la ejecución. [16] Su representación en el diagrama de bloques se muestra en la figura 3.17.

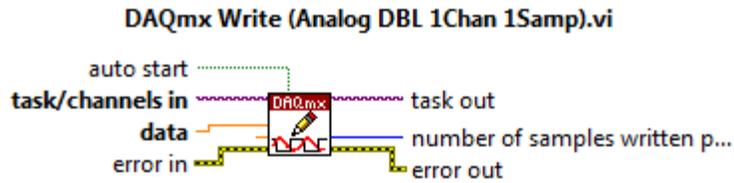


Fig. 3.17: DAQmx Write

En este caso la lista desplegable funciona de la misma forma que en el DAQmx Read, establece el formato de escritura a aplicarse en el canal tal como se indica en la figura 3.18.

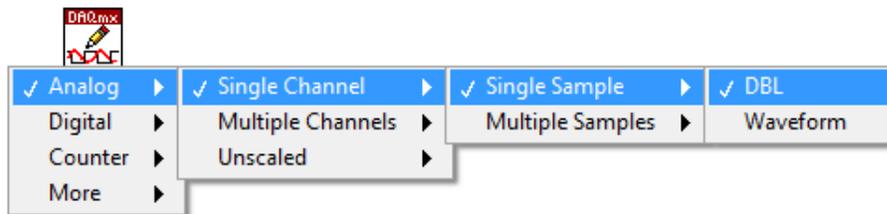


Fig. 3.18: Estableciendo el formato de escritura en el canal

DAQmx Clear Task

Detiene y borra el proceso de adquisición y generación de datos, liberando los recursos software utilizados durante estas etapas. Su representación en el diagrama de bloques se muestra en la figura 3.19.



Fig. 3.19: DAQmx Clear Task

La utilización y configuración de propiedades de estos VI, así como su respectiva conexión en el diagrama de bloques me permiten tener un gran nivel de control sobre la tarjeta. [14]

Observamos en la figura 3.20 la disposición de estos VI para la adquisición de tres señales analógicas a través de un mismo canal. En la figura observamos que se establece un tiempo de muestreo específico de 100 ms por muestra.

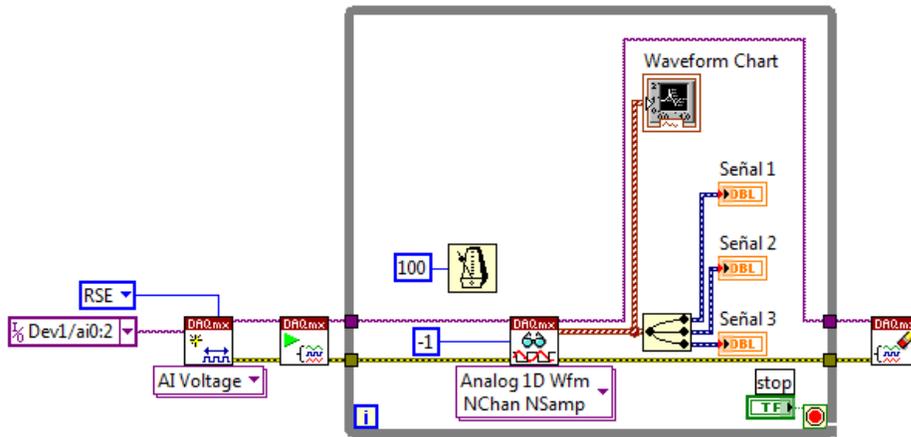


Fig. 3.20: Múltiple adquisición de señales con el NI-DAQmx

En la figura 3.21 observamos la ejecución del código diseñado, efectivamente se está realizando la adquisición de tres señales distintas.

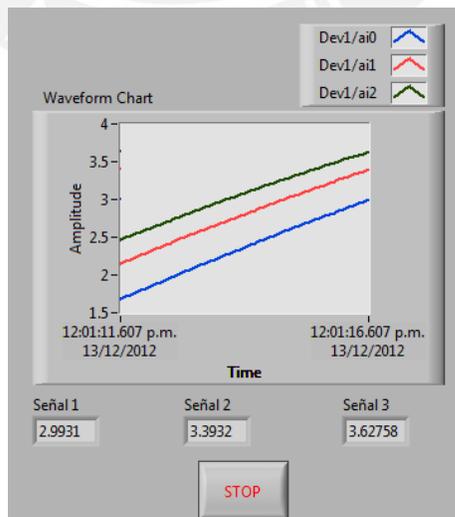


Fig. 3.21: Adquisición de múltiples señales vistas en el panel frontal

De forma similar se procede a configurar los VI descritos, tal como se muestran en la figura 3.22, para la generación de una señal a través de un canal de salida.

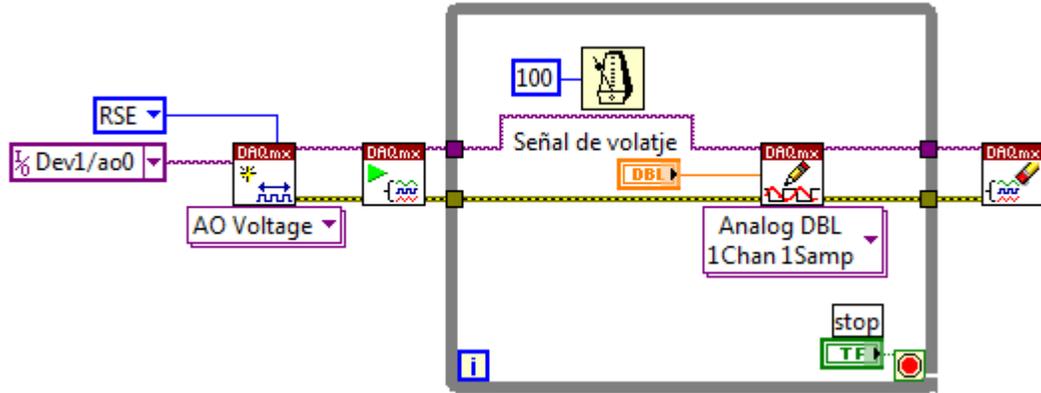


Fig. 3.22: Generación de una señal analógica con el NI-DAQmx

3.4 Escalamiento de señales

El escalamiento consiste en relacionar la señal eléctrica obtenida del sensor de temperatura a un valor requerido [13], en este caso grados centígrados. Se observa en la figura 3.23 el esquema de conexión sobre el cual se está trabajando.

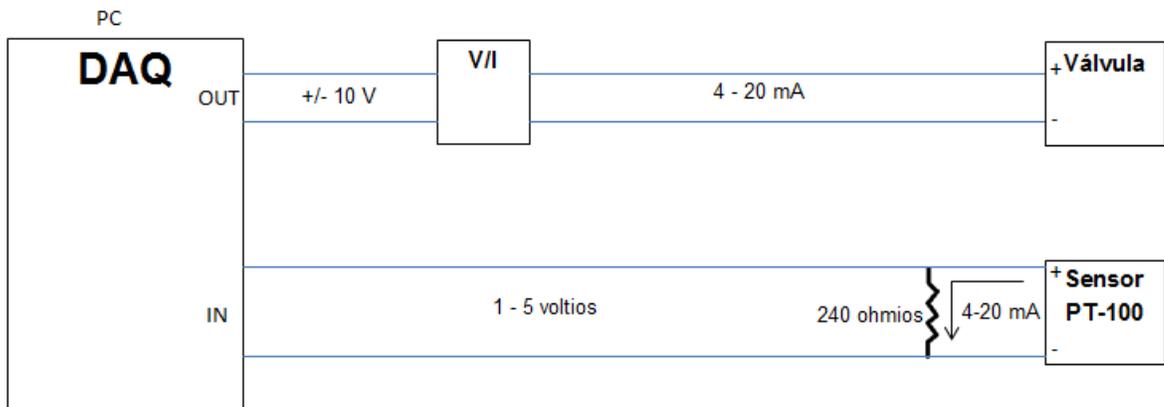


Fig. 3.23: Esquema de conexión

Observamos que la señal que se obtiene del sensor de temperatura varía en un rango de 4 a 20 mA. Se hace circular esta corriente a través de una resistencia

de 240 ohmios, induciendo entre sus terminales un valor de voltaje que será leído por la tarjeta.

Primero se procede a leer valores de voltaje en un canal de la tarjeta para distintos puntos de temperatura cuya medida se realiza a la salida del intercambiador y que va al tanque de drenaje. La lectura de temperatura correspondiente a cada valor de voltaje se hará directamente en la planta debido a que aún no contamos con una ecuación que me relacione estos valores. En la tabla 3.2 se presentan los resultados obtenidos.

T° (C)	V
30.4	1.054
53.7	1.47
58.1	1.61
60.2	1.66
61.3	1.68

Tabla 3.2: Temperatura vs. Voltaje

En la figura 3.24 se plasman los puntos adquiridos en una gráfica y se aproximan a una ecuación lineal que represente en forma muy aproximada el comportamiento del sensor.

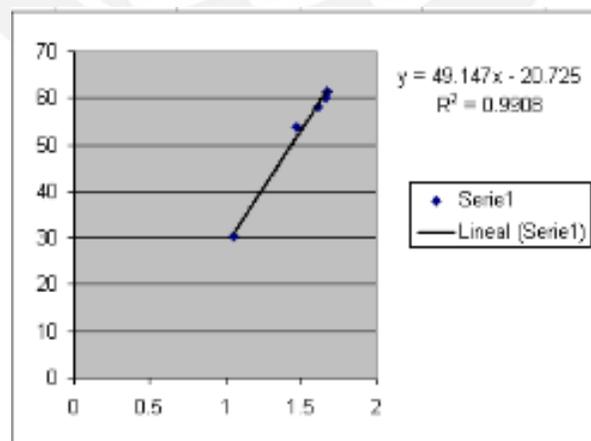


Fig. 3.24: Gráfica T° vs. V y su respectiva ecuación lineal

Observamos que el sensor Pt-100 presenta un comportamiento muy lineal, y podemos aproximar su comportamiento mediante la siguiente ecuación:

$$\text{Temperatura} = 49.147 * \text{Voltaje} - 20.725$$

En la figura 3.25 se observa el código que permite escalar la señal de voltaje adquirida a su valor correspondiente en grados centígrados.

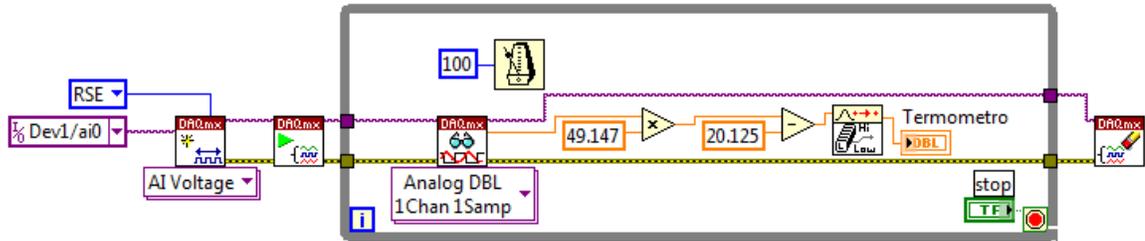


Fig. 3.25: Valor de voltaje escalado a temperatura en C°

Desde la interfaz el usuario podrá determinar de forma interactiva el nivel de apertura de la válvula, se podrá indicar si se desea que este en 0%, 50% o cualquier valor que el operario requiera. Observamos que la tarjeta proporciona una señal de varía entre 10 y -10 voltios destinada a controlar el nivel de apertura previo a su paso por el convertidor de voltaje a corriente. Entonces en el diagrama de bloques se procede a cambiar el rango de operación de (0 – 100%) a (-10 – 10 V). Esto se logra con el código mostrado en la figura 3.26.

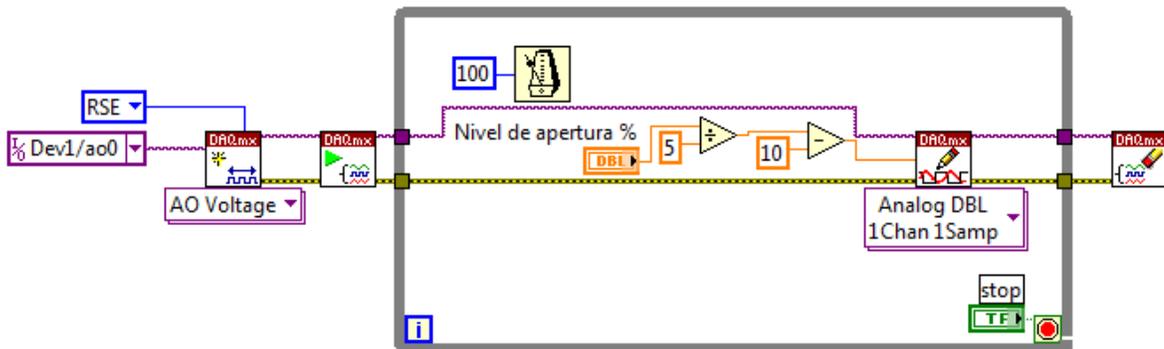


Fig. 3.26: Escalamiento del nivel de apertura de la válvula

Cuando el operario ejecuta el código desde la interfaz, le envía a la válvula una señal que varía entre 10 y -10 voltios, sin embargo la válvula trabaja en un rango de 4 a 20 mA, por lo que se procede a realizar una conversión previa de voltaje a corriente. En la figura 3.27 se muestra el dispositivo que realiza dicha conversión.



Fig. 3.27: Conversor de voltaje a corriente

En la figura 3.28 observamos cómo manualmente se le da a la válvula un nivel de apertura de 75% y como efectivamente en el display ubicado en la misma, se indica que el nivel de apertura corresponde al introducido desde la interfaz.



Fig. 3.28: Control manual de la válvula

3.5 Interfaz del lazo de control

Al momento de diseñar la interfaz se configuran las diferentes señales de entrada y salida sobre la misma. Sobre estas señales se evalúa el

comportamiento de los elementos que hacen parte del lazo de control en contraste con las señales que son enviadas desde el lazo de control.

El lazo de control implica establecer la comunicación de las señales entra la interfaz, los módulos de adquisición de datos y los elementos del lazo de control. [17]

El sistema de control se construye en base a un controlador PID, el cual es un controlador proporcional con acción derivativa y acción integral simultáneamente superpuestas.

Existe un solo conjunto de valores P, I, D que brindan un óptimo rendimiento y encontrarlos requiere: conocimientos teóricos, habilidad obtenida mediante la experiencia y suerte.

En la realidad cualquier conjunto de valores cercano al óptimo brinda un rendimiento aceptable y en algunos casos cercanos al óptimo. Si encontrar los valores que permitan el control óptimo de un proceso se vuelve complicado, se recurren a métodos experimentales para encontrar una aproximación de estos parámetros. [18]

Con el fin de obtener estos parámetros es necesario determinar el rango lineal del proceso. En pruebas realizadas en el laboratorio se pudo determinar que el rango de trabajo lineal para el proceso varía entre 30 y 70% en relación a la variable de control, valor a partir del cual la variable de proceso fue alcanzando su estado estable.

Obtención del modelo matemático

Se utilizó el sistema de control de temperatura en lazo abierto. Usaremos para este fin una señal de prueba tipo escalón la cual será introducida desde la interfaz de LabVIEW dentro del rango lineal del proceso.

Para introducir una señal tipo escalón, la variable de control será modificada desde su valor mínimo hasta su respectivo máximo actuando sobre el nivel de apertura de la válvula desde 0% a 100%. El resultado de aplicar este tipo de señal, la cual se obtuvo empleando la interfaz desarrollada, se muestra en la figura 3.29.

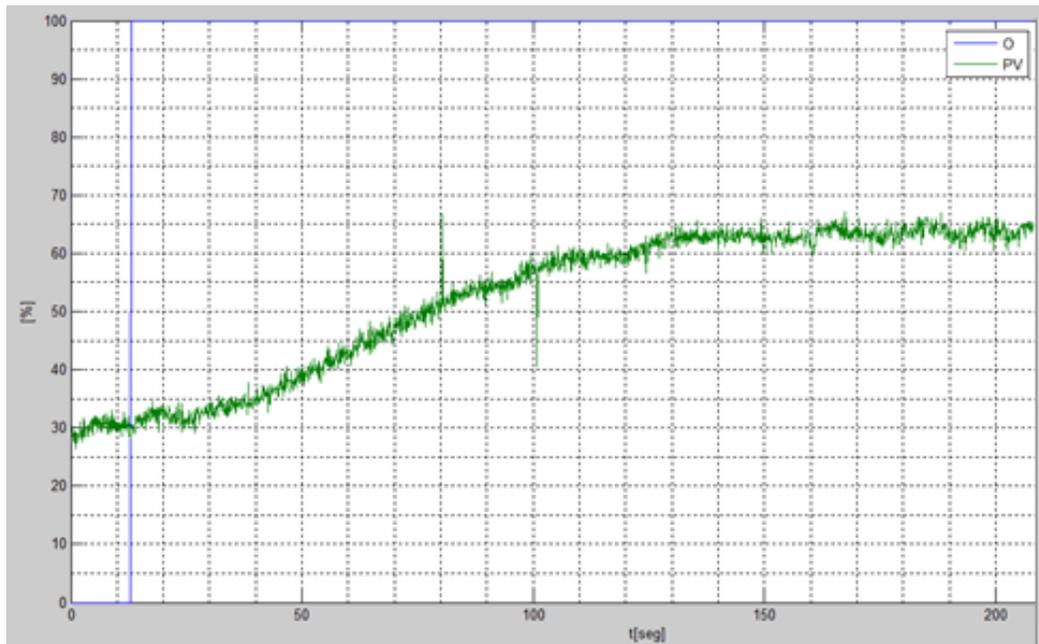


Fig. 3.29: Respuesta en lazo abierto del proceso a una entrada tipo escalón

Una vez obtenida la respuesta recurrimos a modelos experimentales para el diseño del controlador PID. Este proceso se conoce como sintonía del controlador. Por ejemplo se podrá aplicar el método de Zieger & Nichols.

Vemos en la figura 3.29 que la curva de salida exhibe una forma de tipo “S”. Dicha curva está caracterizada por dos parámetros, el tiempo de retardo L y la constante de tiempo T [3]. Basándonos en la gráfica obtenida aproximamos la función de transferencia mediante un sistema de primer orden con un retardo de transporte, el cual se representa mediante la siguiente ecuación:

$$G(s) = \frac{K * e^{-L*s}}{Ts + 1}$$

Ecuación característica de un sistema de primer orden con retardo de transporte

De la gráfica obtenemos los siguientes valores:

- $K = 0.34$
- $L = 15$
- $T = 52$

Con los cuales armamos la función de transferencia de la planta:

$$G(s) = \frac{0.34 * e^{-15s}}{52s + 1}$$

Sintonizando el controlador

Zieger y Nichols propusieron establecer los valores de K_p , T_i y T_d de acuerdo a las expresiones matemáticas que aparecen en la siguiente tabla:

Tipo de Controlador	K_p	T_i	T_d
P	T/L	?	0
PI	$0,9 T/L$	$L/0,3$	0
PID	$1,2 T/L$	$2L$	$0,5L$

Tabla 3.3: Reglas de Zieger & Nichols para la sintonización de controladores

Por lo tanto se obtienen los siguientes parámetros:

$$K_p = 12.23 \quad T_i = 30 \quad T_d = 7.5$$

Es importante recalcar que estos valores pueden no satisfacer el requerimiento de control, sin embargo ofrecen un punto inicial para una sintonización conveniente. [18]

Teniendo estos parámetros como referencia, se procede a su implementación en el entorno de LabVIEW tal y como se muestra en la figura 3.28.

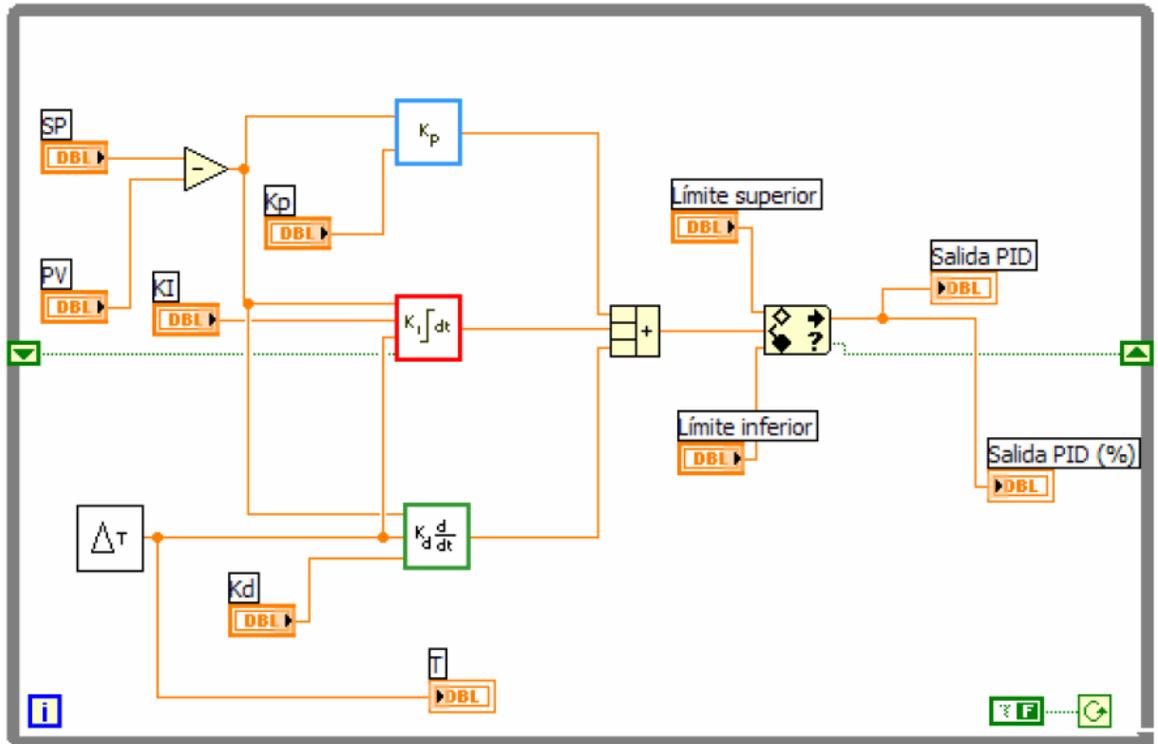


Fig. 3.30: Diagrama de bloques de un controlador PID implementado en LabVIEW

3.6 Diseño del almacenamiento y lectura de datos

En general cuando se ingresan los datos o se obtiene resultados, estos sólo se almacenan en variables de memoria y cuando se apaga la computadora dichos datos se pierden; pero muchas veces algunos datos obtenidos necesitan ser guardados para su posterior recuperación, modificación o análisis, de tal manera que pueda servirnos para la toma de algunas decisiones. LabVIEW nos da la posibilidad de guardar datos importantes en dispositivos de almacenamiento de datos (discos flexibles, disco duro, CDs, etc.). [13]

LabVIEW dispone de una paleta llama File I/O en el cual existen funciones para almacenar datos, los cuales se observan en la figura 3.31.

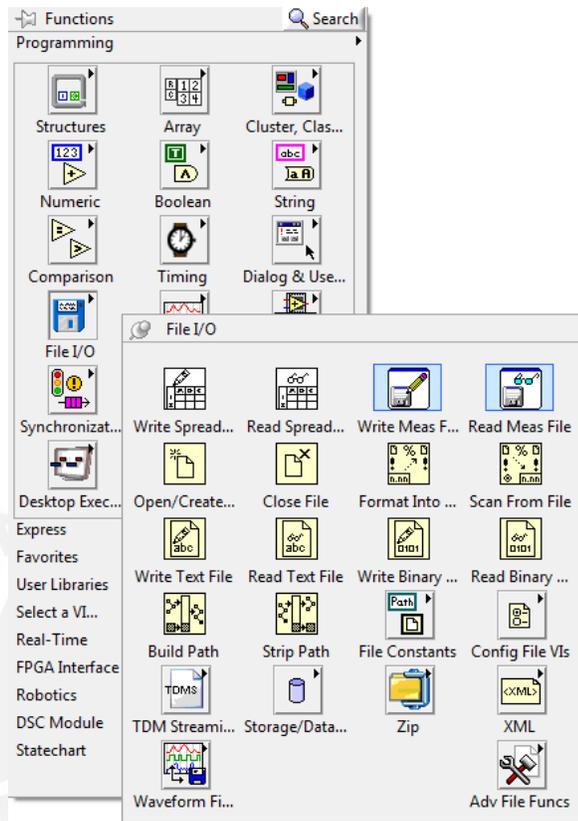


Fig. 3.31: Paleta de funciones para el almacenamiento y lectura de archivos

Almacenamiento de datos en un archivo nuevo

Para guardar datos en un archivo se han de seguir los siguientes tres pasos: crear el archivo, escribir los datos en el archivo y cerrar el archivo. [16]

Para cumplir con este requerimiento usaremos los siguientes bloques:

Open/Create/Replace File

En la figura 3.32 se muestra la representación de esta función en el diagrama de bloques. Su función es crear o abrir un archivo o reemplazar un archivo existente. Si no se indica de forma explícita la ubicación del archivo aparecerá una ventana especificando su ruta [16]. Se debe indicar también la operación a realizar (abrir, crear o reemplazar).

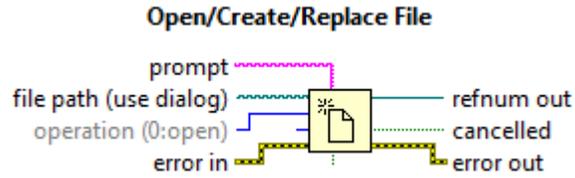


Fig. 3.32: Función Open/Create/Replace File

Write to text file:

Se encarga de la escritura de caracteres como líneas para un archivo. Su representación se muestra en la figura 3.33.

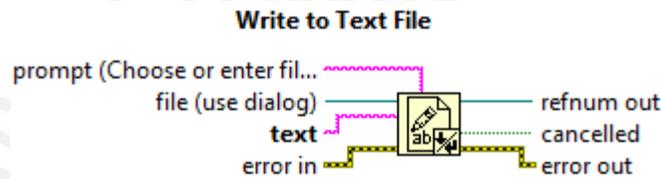


Fig. 3.33: Función Write to Text File

Close file

La función de este bloque es liberar el archivo una vez terminado el proceso de escritura. Su representación se muestra en la figura 3.34.

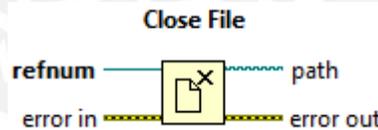


Fig. 3.34: Función Close File

Number to Fractional String

El trabajo propuesto implica almacenar datos numéricos obtenidos en la parte de adquisición. La inserción de dichos datos en un archivo implica la conversión de los mismos a una cadena de caracteres [13]. La función Number to Fractional String, mostrada en la figura 3.35, permite esta conversión, permitiendo precisar el número de caracteres, así como el número de decimales que conformarán la cadena de caracteres generada.

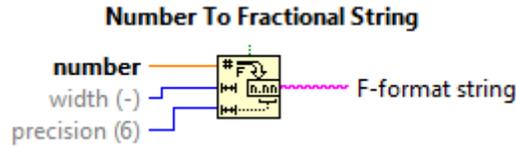


Fig. 3.35: Función Number to Fractional String

Build Path

Para la creación del archivo o carpeta que se quiere generar es necesario que se especifique la ruta de almacenamiento. La función Build Path, mostrada en la figura 3.36, permite al usuario especificar dicha ruta, así como asignarle un nombre al archivo generado. Se debe especificar al final del nombre el tipo de archivo que se desea generar (.txt para un archivo de texto o .xls para un archivo en Excel). [13]

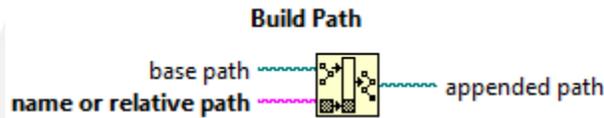


Fig. 3.36: Función Build Path

Create Folder

Función que permite la creación de la carpeta en la cual estarán contenidos los archivos creados. En la figura 3.37 se muestra su representación. Es necesario especificar la ruta de creación, así como el nombre de la carpeta. El formato más utilizado al momento de asignar el nombre a una carpeta es la fecha del sistema. [16]

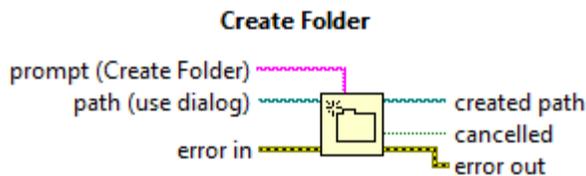


Fig. 3.37: Función Create Folder

Para acceder a la fecha y hora del sistema se hace uso de la función Format Date/Time String, mostrada en la figura 3.38.

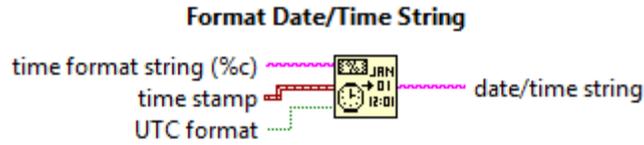


Fig. 3.38: Función Date/Time String

En la figura 3.39 se observa la disposición de estas funciones en el diagrama de bloques que hará posible la generación de carpetas y archivos que permitirán el almacenamiento de la data que el usuario quiera almacenar. En la interfaz se proporciona la opción de iniciar o detener el almacenamiento de datos mediante la activación de un interruptor ubicado en el panel frontal.

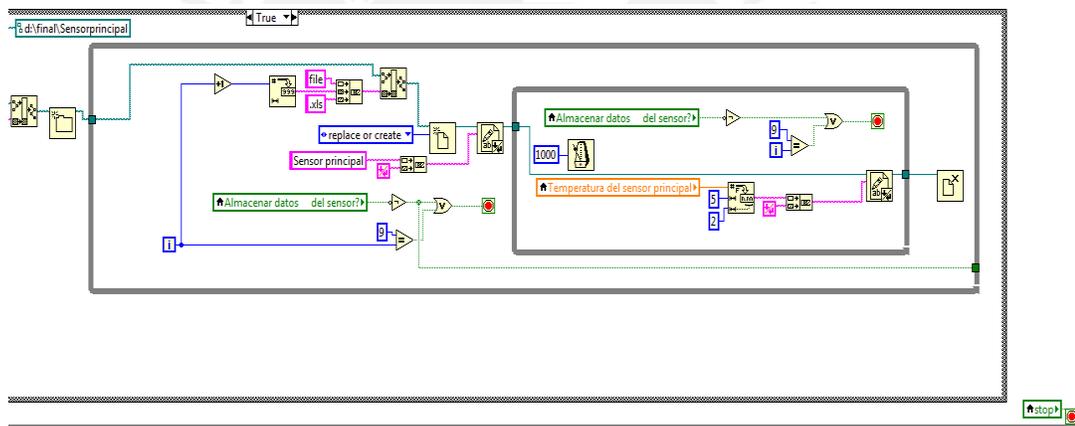


Fig. 3.39: Almacenamiento de datos en LabVIEW

Lectura de un archivo

Para la lectura de datos se utilizan, adicionalmente a los bloques que se han descrito, las siguientes funciones:

Read from Text File

Esta función se encarga de leer un número determinado de caracteres o líneas de un archivo. Su representación se muestra en la figura 3.40.

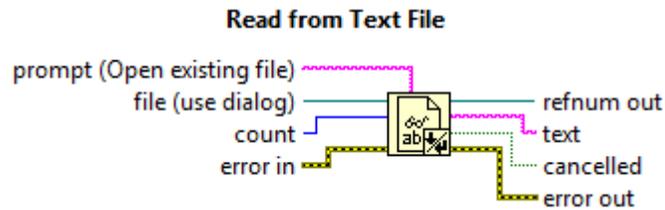


Fig. 3.40: Función Read from Text File

String Subset

Permite extraer una cadena de caracteres a partir de un desfase especificado. Su representación se muestra en la figura 3.41.

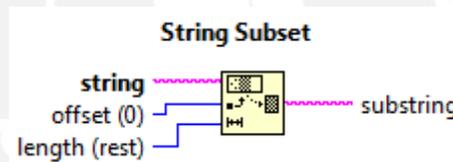


Fig. 3.41: Función String Subset

Spread-sheet String to Array

Almacena en un arreglo numérico los valores extraídos del archivo texto. Su representación se muestra en la figura 3.42.

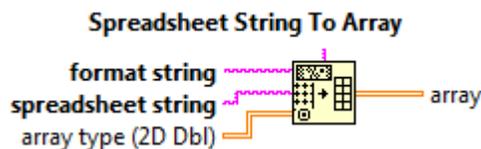


Fig. 3.42: Función Spreadsheet String To Array

Lectura por sensor

Este subVI creado muestra en un indicador de caracteres los valores de la variable sensada. Su representación se muestra en la figura 3.43.

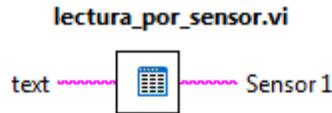


Fig.3.43: Sub-VI Lectura por sensor

En la figura 3.44 se observa la disposición final de estas funciones en el diagrama de bloques que hará posible la lectura de datos almacenados y respectiva visualización en el panel frontal.

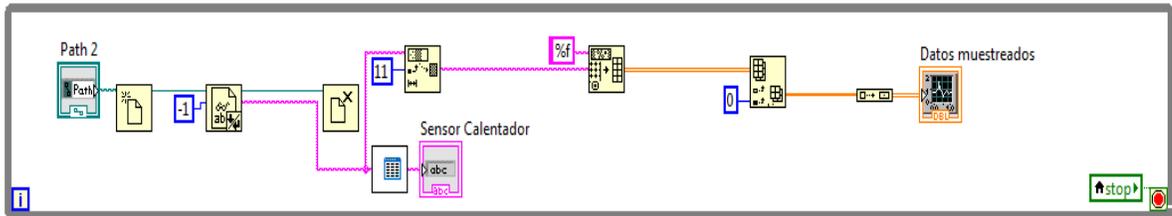


Fig. 3.44: Lectura de datos en LabVIEW

3.7 Comunicación MATLAB - LABVIEW

El entorno de programación de LabVIEW ofrece una función llamada “MATLAB Script Node”. Sobre la cual se pueden escribir funciones propias del entorno MATLAB y aplicarlas a variables definidas desde LabVIEW [13]. En la figura 3.45 se observa la estructura de esta función.

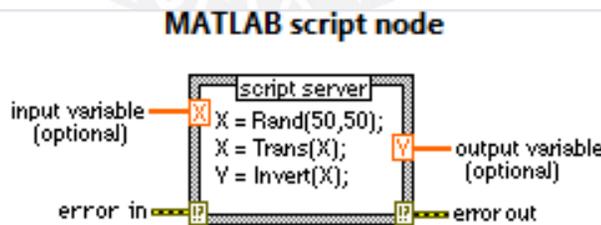


Fig. 3.45: Función MATLAB script node

Sobre dicha estructura se definen variables de entrada para las funciones propias de MATLAB y variables de salida las cuales serán usadas en el entorno LabVIEW.

El propósito de usar esta función es el de comunicar el entorno LABVIEW a una función de detección de fallos para válvulas (ver tesis de Yoshimar Rodríguez Donayre), basado en el método de Horch, desarrollado en forma paralela al presente trabajo de estudio.

Dicho algoritmo requiere del almacenamiento de 4000 datos de la variable de proceso y 4000 datos de la variable de control. Razón por la cual se ubica en el panel frontal un interruptor para iniciar dicha captura de datos. Así como también, se proporciona otro interruptor para la ejecución del mismo.

En la figura 3.46 observamos el código que permite la ejecución del algoritmo cuando el usuario lo crea pertinente.

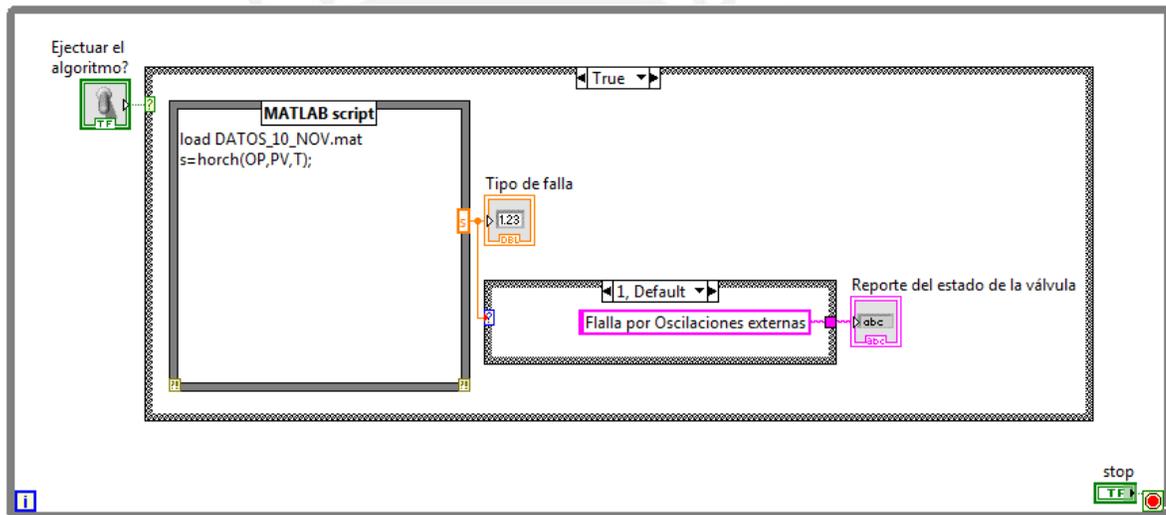


Fig. 3.46: Ejecución en LabVIEW de un algoritmo desarrollado en MATLAB

3.8 Comunicación LabVIEW - Simulink

La comunicación entre la interfaz de usuario de LabVIEW con un modelo de simulación ejecutado en Simulink se da a través de una herramienta llamada SIT (Simulation Interface Toolkit).

Esta herramienta permite manipular variables definidas, así como visualizar el resultado de funciones ejecutadas en el entorno Simulink. Es indispensable que

ambos programas se encuentren en ejecución al mismo tiempo. La comunicación entre ambos entornos se produce gracias a un servidor propio de la herramienta denominado “SIT Server”. El cual entabla una comunicación TCP/IP a través del puerto 6011 sobre el cual se hará la transferencia de datos. [13]

Se requiere de esta herramienta debido a que el algoritmo para la detección de bombas (ver tesis de Anderson Barboza Cerván), desarrollado en forma paralela al presente trabajo de estudio, ha sido desarrollado en el entorno Simulink.

En la figura 3.47 se observa el algoritmo para la detección de fallos en válvulas. El bloque “Signal Probe” ubicado en la parte superior del modelo es el que permite la comunicación entre el Simulink y LabVIEW.

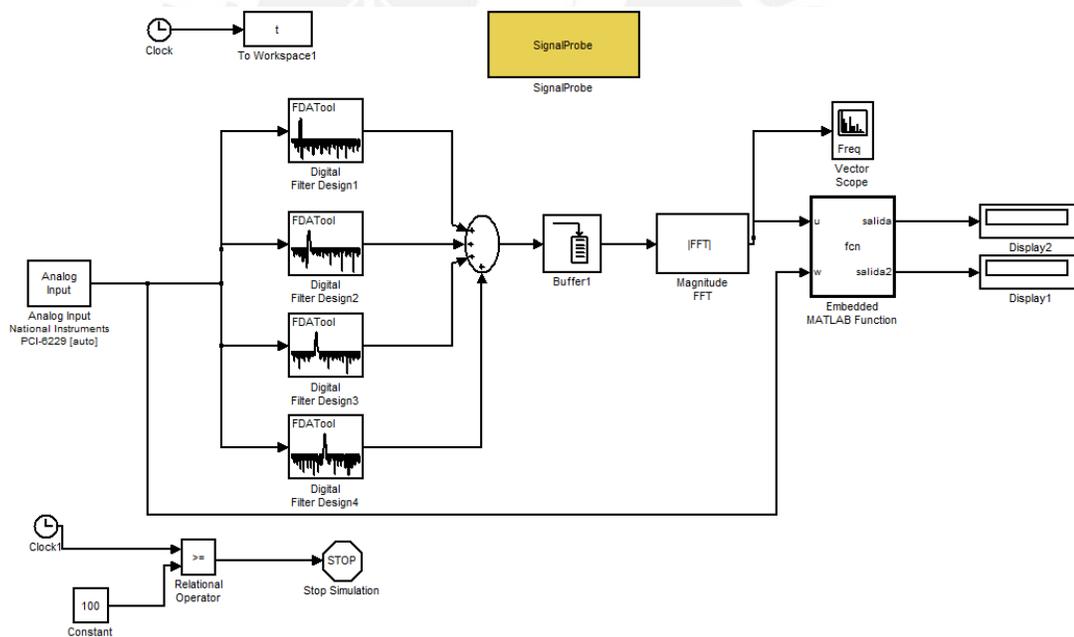


Fig. 3.47: Modelo para la detección de fallos en bombas con Signal Probe

La comunicación entre los programas se hace en LabVIEW a través del SIT Connection Manager. Herramienta a través de la cual determino el modelo de Simulink, así como el mapeo entre las variables que van a comunicarse.

En la figura 3.48 se aprecia el mapeo de la variable en LabVIEW que almacenará el tipo de falla y el display que almacenará el tipo de falla en Simulink.

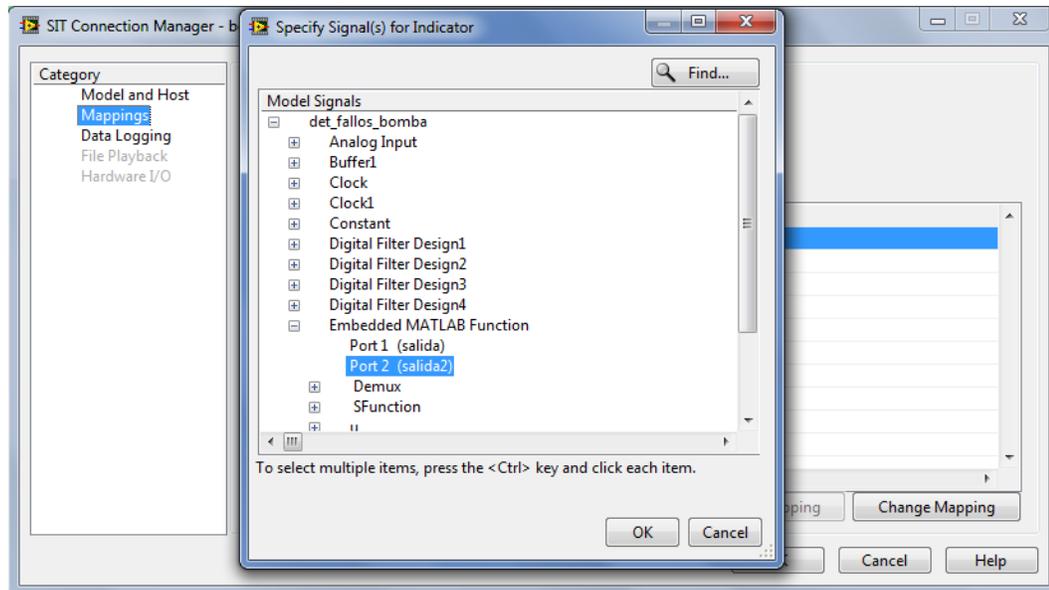


Fig. 3.48: Mapeo de variables entre Simulink y LabVIEW

Una vez configurada la comunicación en LabVIEW, aparece en el panel frontal una paleta de funciones que permiten ejecutar el programa en Simulink desde LabVIEW.

3.9 Diseño final de la interfaz

Contemplando los criterios de diseño explicados hasta este punto, apreciamos en la figura 3.49 la interfaz de usuario desarrollada en LabVIEW.

En la misma figura se observa cómo están dispuestas en pestañas las funciones que involucran los objetivos de este proyecto.

El resultado y análisis de explorar las pestañas restantes se presentan en el capítulo 4.

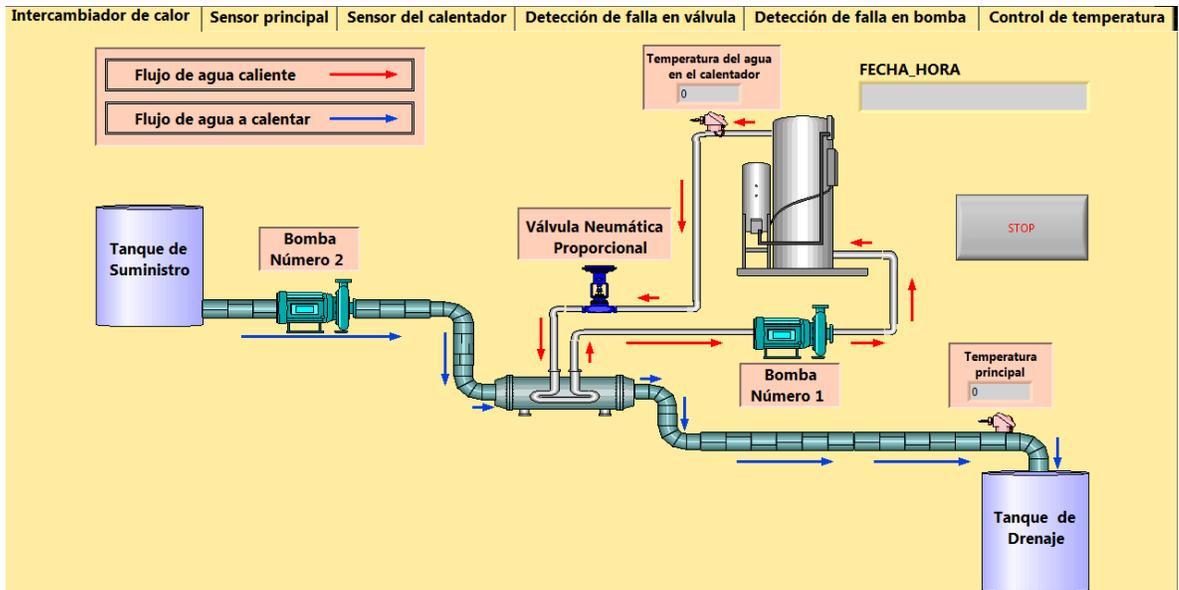


Fig. 3.49: Interfaz para la planta intercambiador de calor

En conclusión, con el desarrollo del trabajo propuesto se puede integrar la interfaz hombre-máquina con:

- El sistema de detección de fallos en la válvula de la planta.
- El sistema de detección de fallos en las bombas de la planta.

CAPÍTULO 4

PRUEBAS Y ANÁLISIS DE RESULTADOS

4.1 Intercambiador de calor

Se presenta en la figura 4.1 la interfaz de usuario final que plasma en pantalla la planta objeto de estudio en funcionamiento.

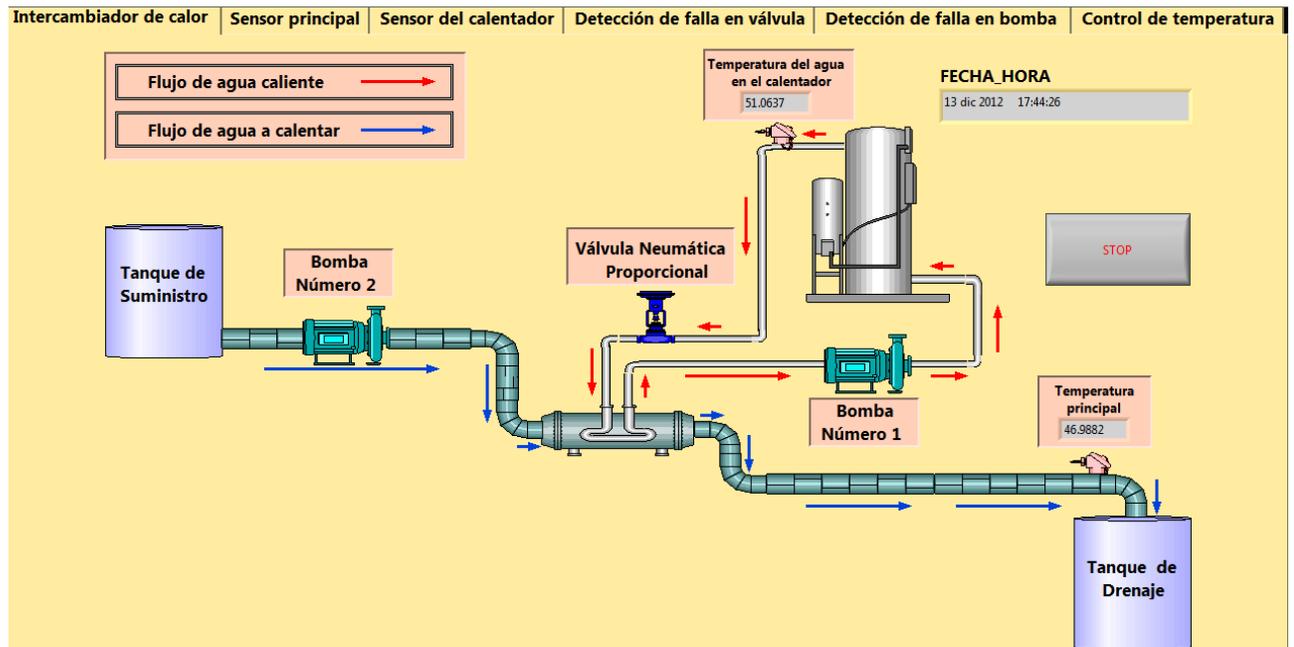


Fig. 4.1: Interfaz para la planta intercambiador de calor

En la figura 4.1 también observamos, en base a los criterios de diseño detallados en el capítulo tres, los indicadores de temperatura en funcionamiento. El usuario podrá monitorear desde la pantalla el valor de la variable de proceso y el valor de la temperatura en el calentador sin necesidad de acercarse a la planta.

4.2 Almacenamiento y lectura de datos de la variable de proceso

En la figura 4.2 se muestra la interfaz con la cual el usuario podrá observar la evolución de la variable de proceso en tiempo real a través de un gráfico

indicador. También se observa el interruptor con el cual se inicia el almacenamiento de datos.

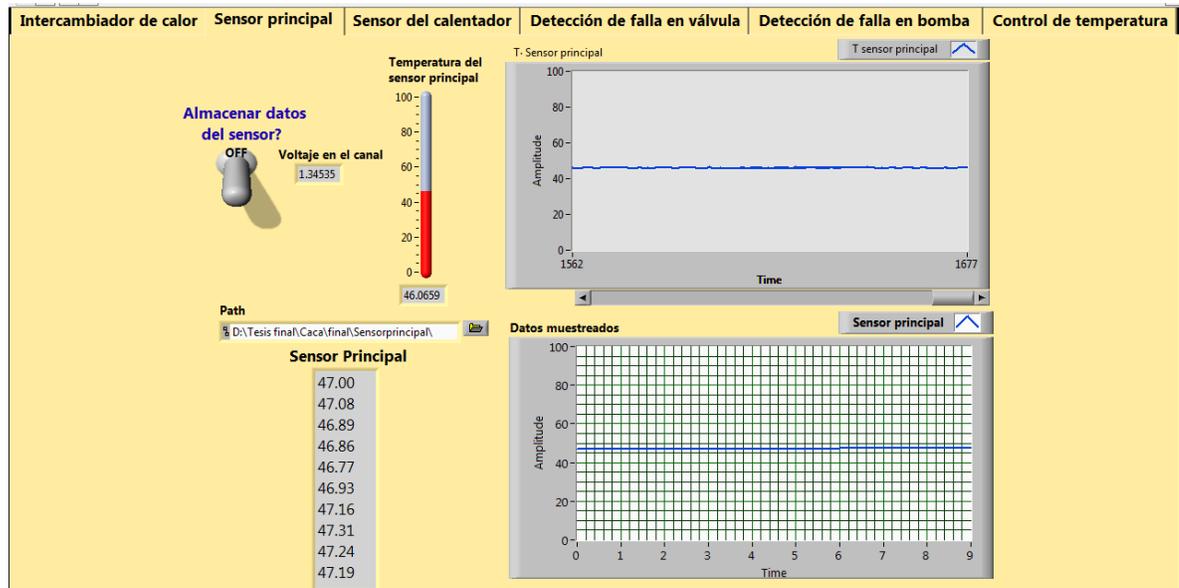


Fig. 4.2: Interfaz de monitoreo para la variable de proceso

En la figura 4.2 se observa como en la gráfica superior se muestra el valor de la variable de proceso inmediatamente después de su adquisición.

En la figura 4.3 se aprecia la activación del interruptor que permite el almacenamiento de datos de la variable de proceso.



Fig. 4.3: Interruptor para almacenar datos de la variable de proceso

En la figura 4.4 se observa que inmediatamente después que se activa el interruptor, se crea en la carpeta especificada otra carpeta en cuyo nombre se especifica el día, hora y fecha que se está haciendo el registro.

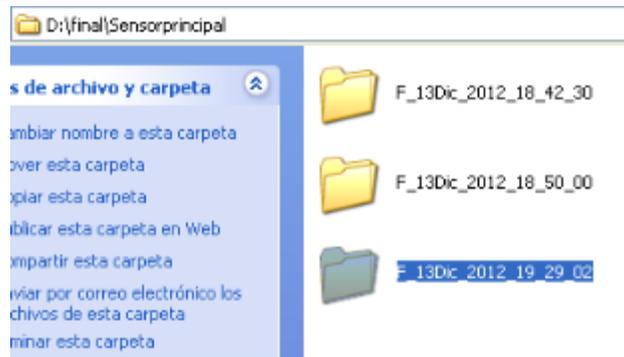


Fig. 4.4: Carpeta de almacenamiento para la variable de proceso

En la figura 4.4 se observa en la parte superior la ruta especificada para el almacenamiento.

En la figura 4.5 se observan los archivos contenidos en la carpeta creada. El acceso a dichos archivos se realiza desde la interfaz.

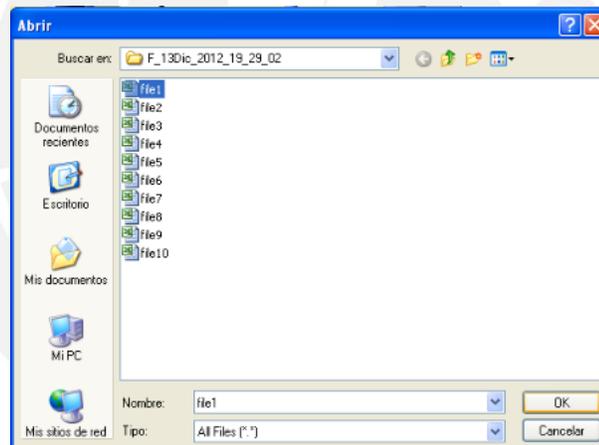
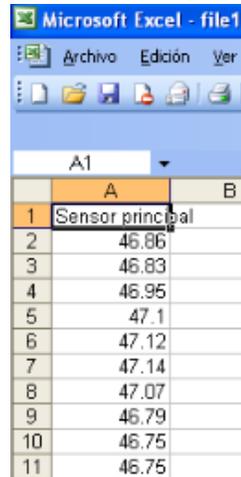


Fig.4.5: Archivos contenidos en la carpeta creada desde la interfaz

En la figura 4.6 se puede apreciar en detalle los valores contenidos en el primer archivo, los cuales podrán ser vistos y graficados por el usuario.

En la figura 4.7 se observan los valores de la variable de proceso con su respectiva gráfica. Notar que los valores almacenados en el archivo seleccionado son iguales a los valores mostrados en la interfaz.



	A	B
1	Sensor principal	
2	46.86	
3	46.83	
4	46.95	
5	47.1	
6	47.12	
7	47.14	
8	47.07	
9	46.79	
10	46.75	
11	46.75	

Fig. 4.6: Valores de la variable de proceso en un archivo

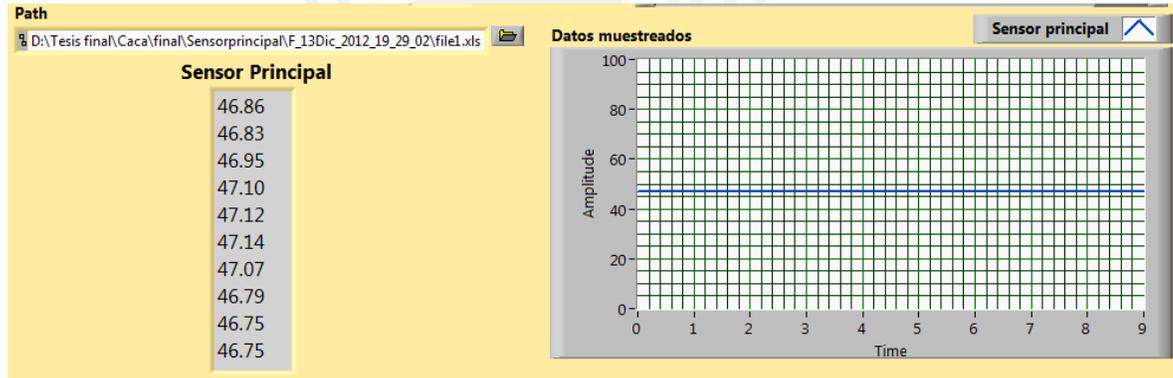


Fig. 4.7: Lectura y gráfica de datos de la variable de proceso almacenados

4.3 Almacenamiento y lectura de datos de la temperatura en el calentador

En la figura 4.8 se muestra la interfaz con la cual el usuario podrá observar la evolución de la temperatura en el calentador en tiempo real a través de un gráfico indicador. También se observa el interruptor con el cual se inicia el almacenamiento de datos.

En la misma figura se observa como en la gráfica superior se muestra el valor de la variable de proceso inmediatamente después de su adquisición.

En la figura 4.9 se aprecia la activación del interruptor que permite el inicio del almacenamiento de datos de la temperatura en el calentador.

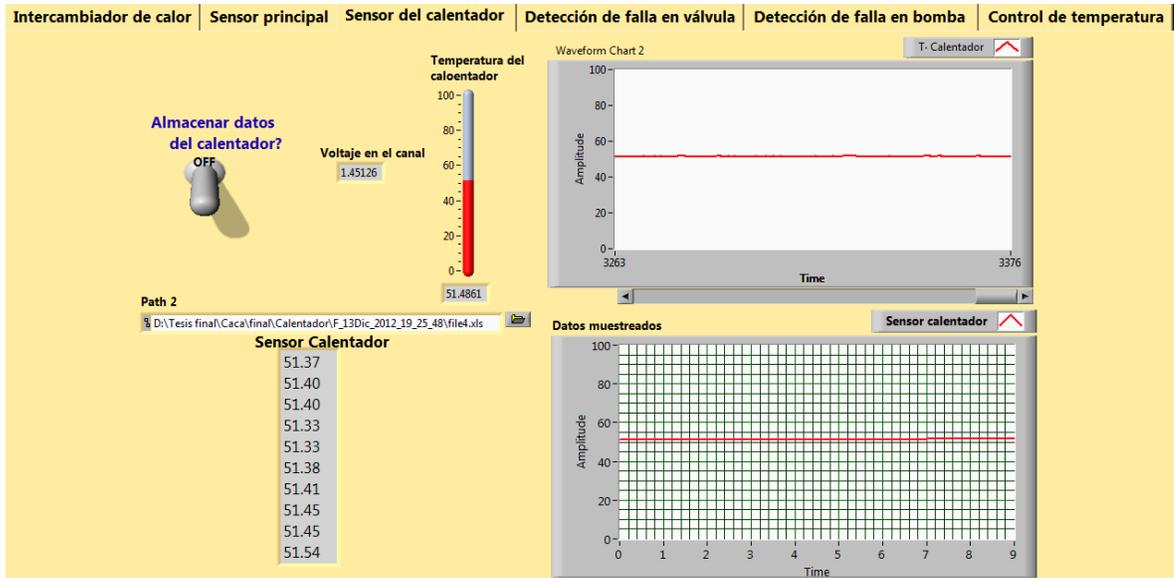


Fig. 4.8: Interfaz de monitoreo para la temperatura en el calentador



Fig. 4.9: Interruptor para almacenar datos de la temperatura del intercambiador

En la figura 4.10 se observa que inmediatamente después que se activa el interruptor, se crea en la carpeta especificada otra carpeta en cuyo nombre se especifica el día, hora y fecha que se está haciendo el registro.

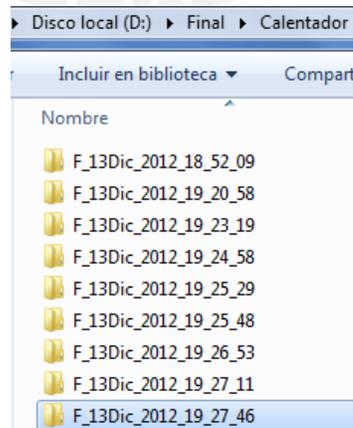


Fig. 4.10: Carpeta de almacenamiento para la temperatura en el calentador

En la figura 4.10 se observa en la parte superior la ruta especificada para el almacenamiento.

En la figura 4.11 se observan los archivos contenidos en la carpeta creada. El acceso a dichos archivos se realiza desde la interfaz.

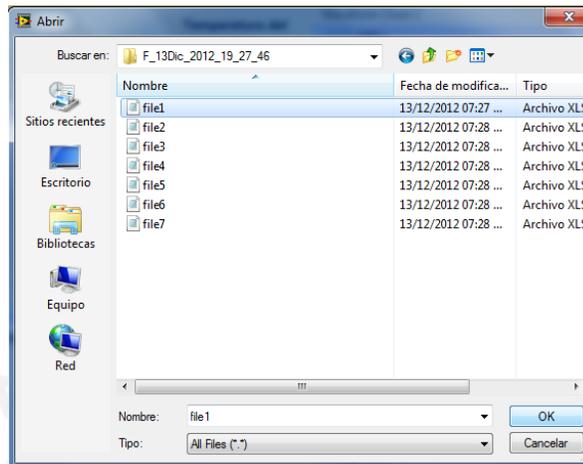


Fig. 4.11: Archivos contenidos en la carpeta creada desde la interfaz

En la figura 4.12 se puede apreciar en detalle los valores contenidos en el primer archivo, los cuales podrán ser vistos y graficados por el usuario.

A1	
	A
1	Calentador
2	51.15
3	51.4
4	51.78
5	51.78
6	51.7
7	51.47
8	51.36
9	51.31
10	51.36
11	51.4

Fig. 4.12: Valores de la temperatura del calentador en un archivo

En la figura 4.13 se observan los valores de la temperatura del calentador con su respectiva gráfica. Notar que los valores almacenados en el archivo seleccionado son iguales a los valores mostrados en la interfaz.

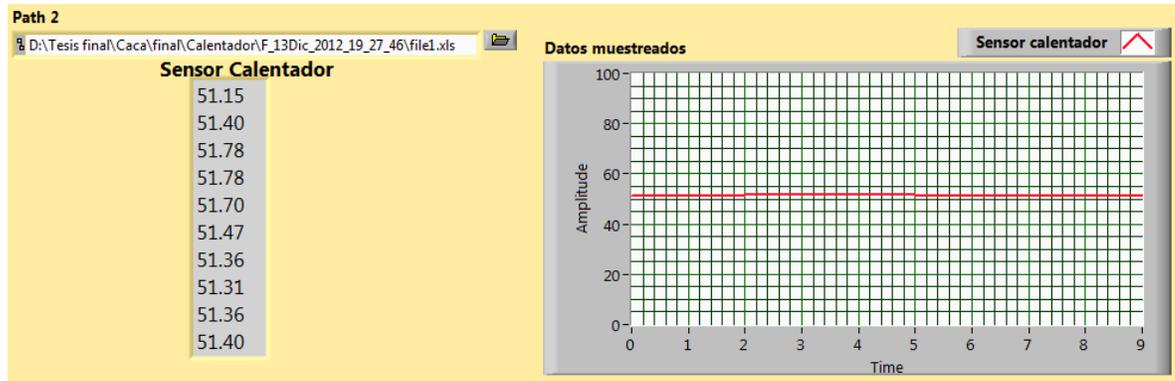


Fig. 4.13: Lectura y gráfica de datos de la temperatura del calentador almacenados

4.4 Detección de fallos en la válvula

En la figura 4.12 se observa la interfaz a través de la cual el usuario interactuará con el algoritmo de detección de fallos para válvulas desarrollado en MATLAB.

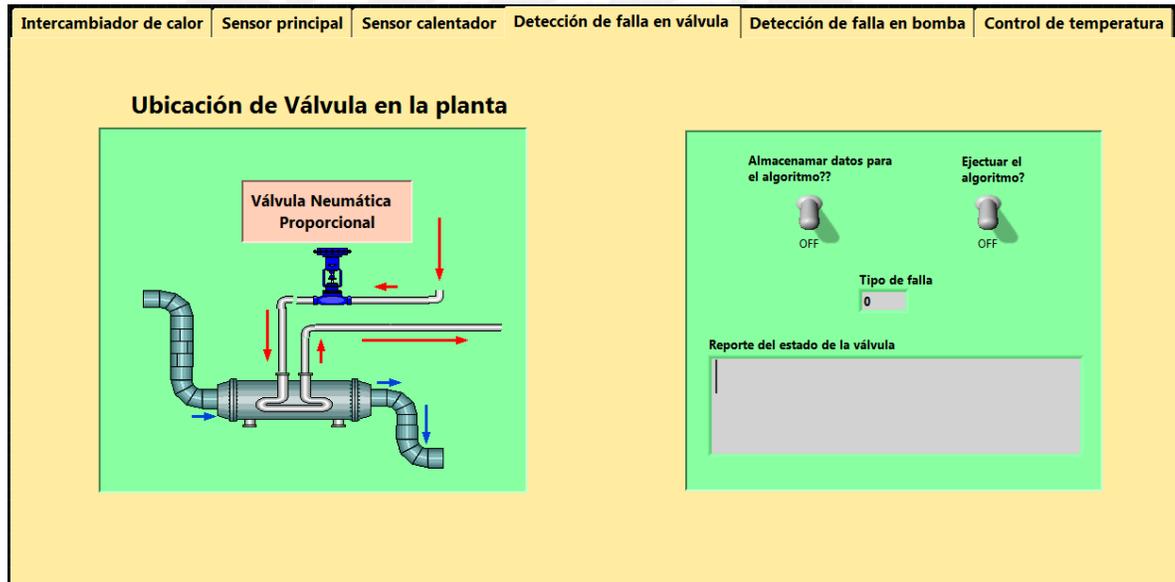


Fig. 4.12: Interfaz de usuario para la detección de fallos en la válvula

En la figura 4.13 se aprecia la activación del interruptor para iniciar el almacenamiento de datos necesario para la ejecución del algoritmo.

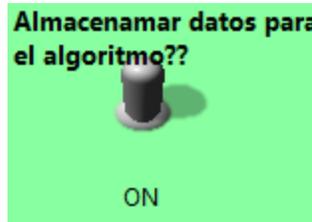


Fig. 4.13: Interruptor para almacenar datos para el algoritmo

En la figura 4.14 se observa la generación del archivo que contendrá la información necesaria para la ejecución del algoritmo.

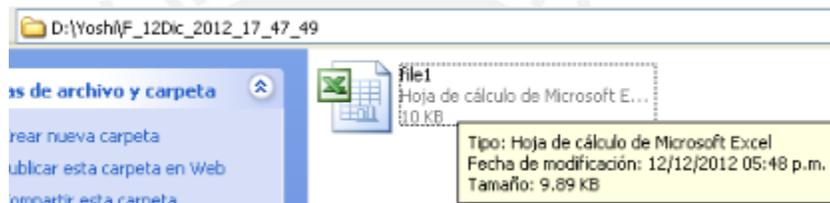


Fig. 4.14: Generación del archivo

En la figura 4.15 se detallan los datos que necesita el algoritmo para su ejecución en MATLAB.

	A	B	C
1	SP	PV	OP
2	60	60.22	52.94
3	60	61.82	52.94
4	60	61.82	54.03
5	60	59.1	51.75
6	60	63.18	55.23
7	60	60.26	52.33
		⋮	
3992	60	59.79	71.71
3993	60	59.6	71.43
3994	60	58.9	70.97
3995	60	62.21	73.47
3996	60	60.47	71.56
3997	60	60.62	72.02
3998	60	61.68	72.74
3999	60	59.72	71.13
4000	60	61.31	72.67
4001	60	62	72.83

Fig. 4.15: Datos para la ejecución del algoritmo

Cuando se tiene el archivo almacenado, se procede a cargar los datos en el workspace de MATLAB, así como también el algoritmo de detección desarrollado en forma paralela al presente trabajo de estudio.

Cargado los datos, el algoritmo está listo para ser ejecutado desde la interfaz. En la figura 4.16 se observa el resultado de ejecutar el algoritmo.

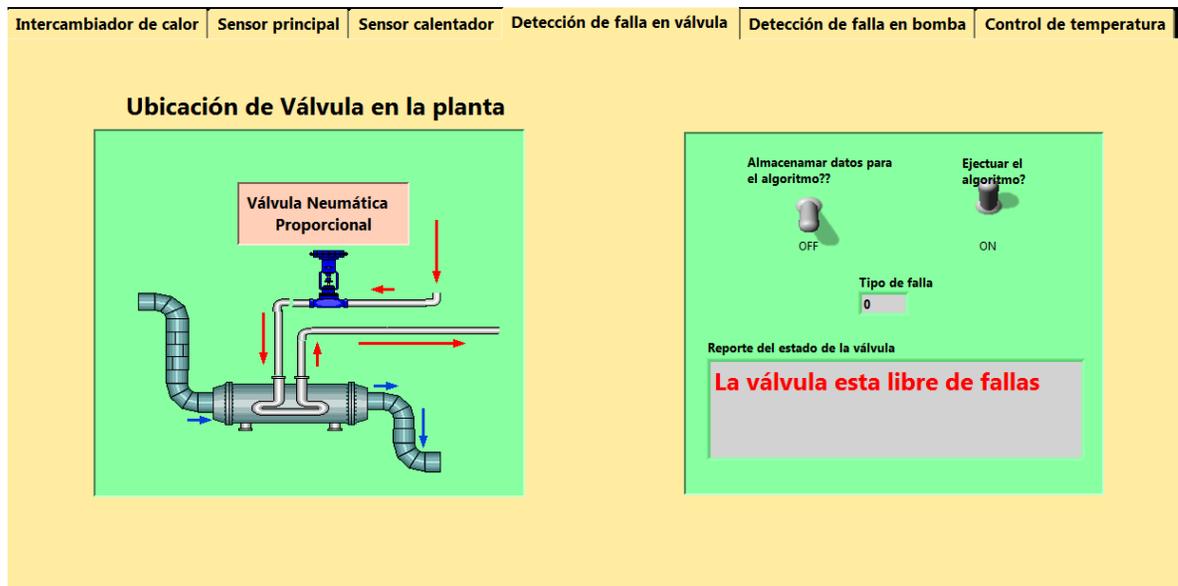


Fig. 4.16: Resultado de ejecutar el algoritmo de detección de fallos para válvulas desde la interfaz

4.5 Detección de fallos en la bomba

En la figura 4.17 se aprecia la interfaz con la cual el usuario interactuará con el algoritmo de detección de fallos desarrollado en Simulink.

El algoritmo será aplicado a la bomba sobre la cual se encuentre el acelerómetro. La comunicación entre los entornos de LabVIEW y Simulink ya se encuentra establecida, pues como vemos en la figura 4.17 ya se dispone de la paleta que permite la ejecución del algoritmo desde la interfaz.

En la figura 4.18 se muestra el resultado de ejecutar el algoritmo, se observa también el cambio en el estado de ejecución, el cual cambió de “Stopped” (detenido) a “Running” (en ejecución).

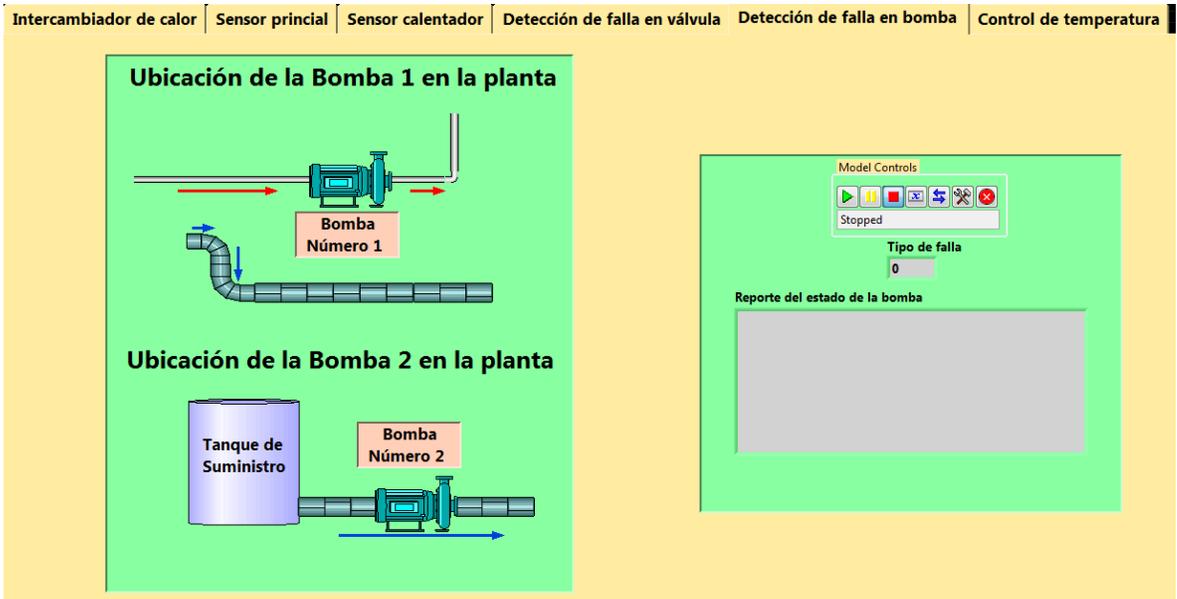


Fig. 4.17: Interfaz de usuario para la detección de fallos en la bomba

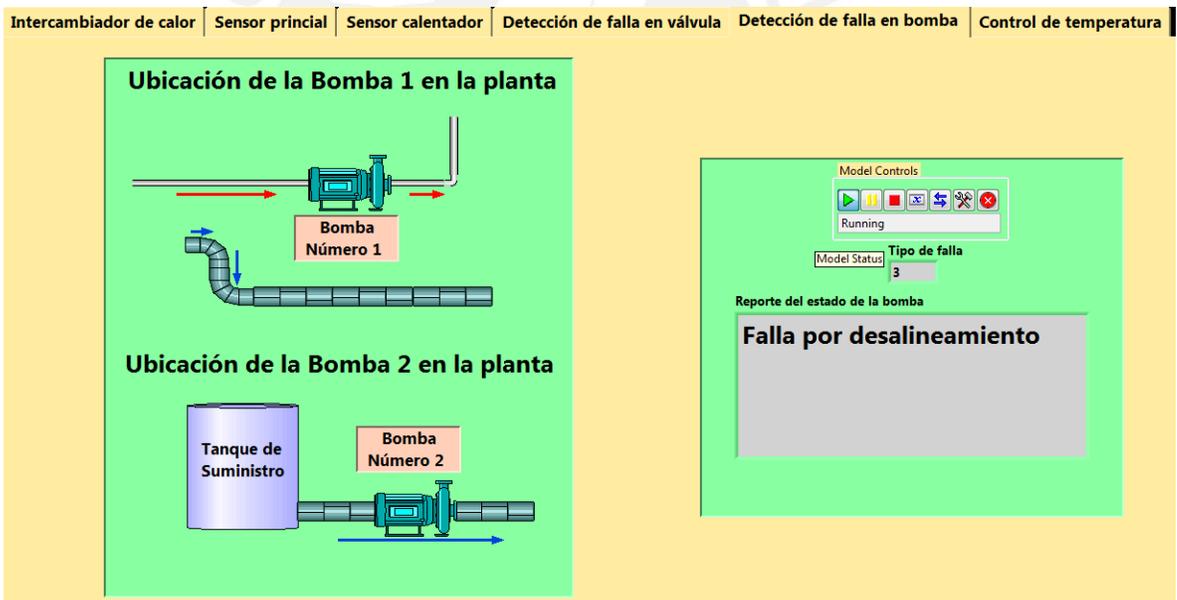


Fig. 4.18: Resultado de ejecutar el algoritmo de detección de fallos para bombas desde la interfaz

4.6 Interfaz de control

Se toma como punto inicial los parámetros K_p , T_i y T_d hallados con el método de Ziegler & Nichols. A partir de estos valores se procede a sintonizar el controlador desde la interfaz de LabVIEW.

En la figura 4.19 observamos la interfaz a través de la cual el usuario podrá interactuar con el sistema de control. En dicha interfaz se proporciona al usuario la capacidad para determinar a través de la activación de un interruptor si desea trabajar el sistema de control en modo automático o manual.

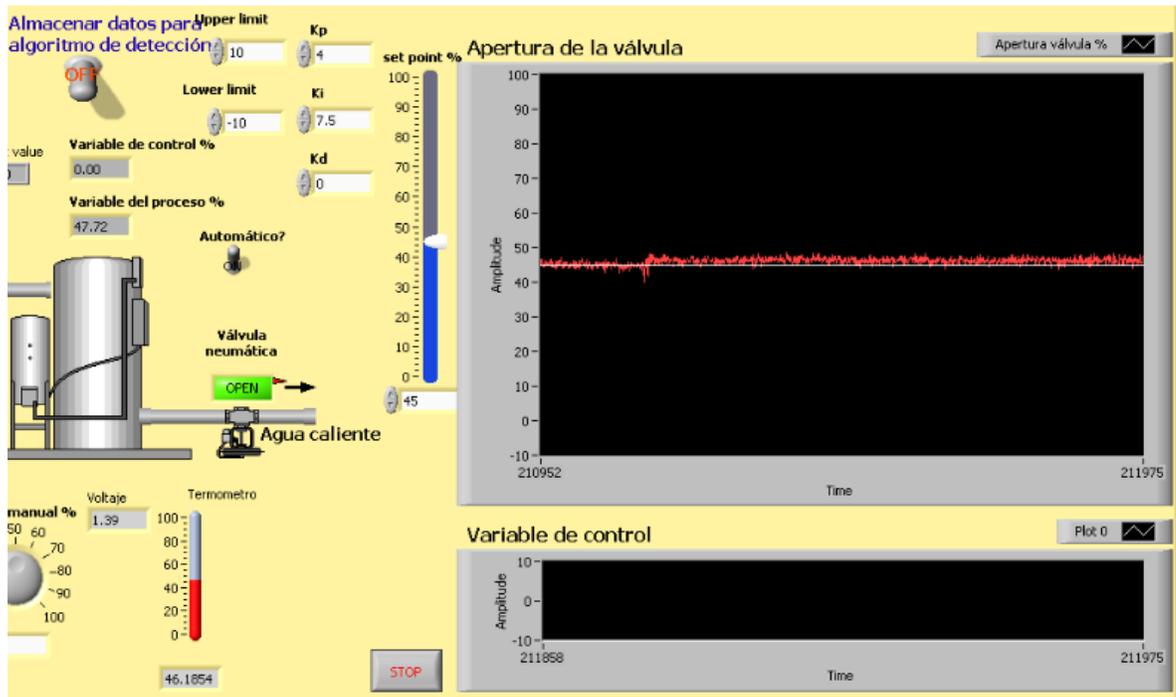


Fig. 4.19: Interfaz de control implementado en LabVIEW

En la figura 4.19 se observa cómo es posible sintonizar el controlador desde la pantalla de un computador. Se aprecia también como es que el valor de la variable de proceso se aproxima al valor de set point establecido también en la interfaz.

Si el usuario varía el set point a un valor más elevado, la acción correctiva que toma el sistema es de abrir la válvula para que más flujo de agua caliente transite hacia el intercambiador. Esta acción correctiva se observa en la figura 4.20, en donde se apreció que al variar el set point a un punto inferior, la válvula permanece cerrada esperando el intercambio de calor apropiado para disminuir la temperatura. Sin embargo, en la misma figura se aprecia también que al incrementar el set point, efectivamente el nivel de apertura de la válvula representado por la variable de control empieza a aumentar, resultando también

en el incremento de flujo de agua caliente y en consecuencia el incremento de la variable de proceso.

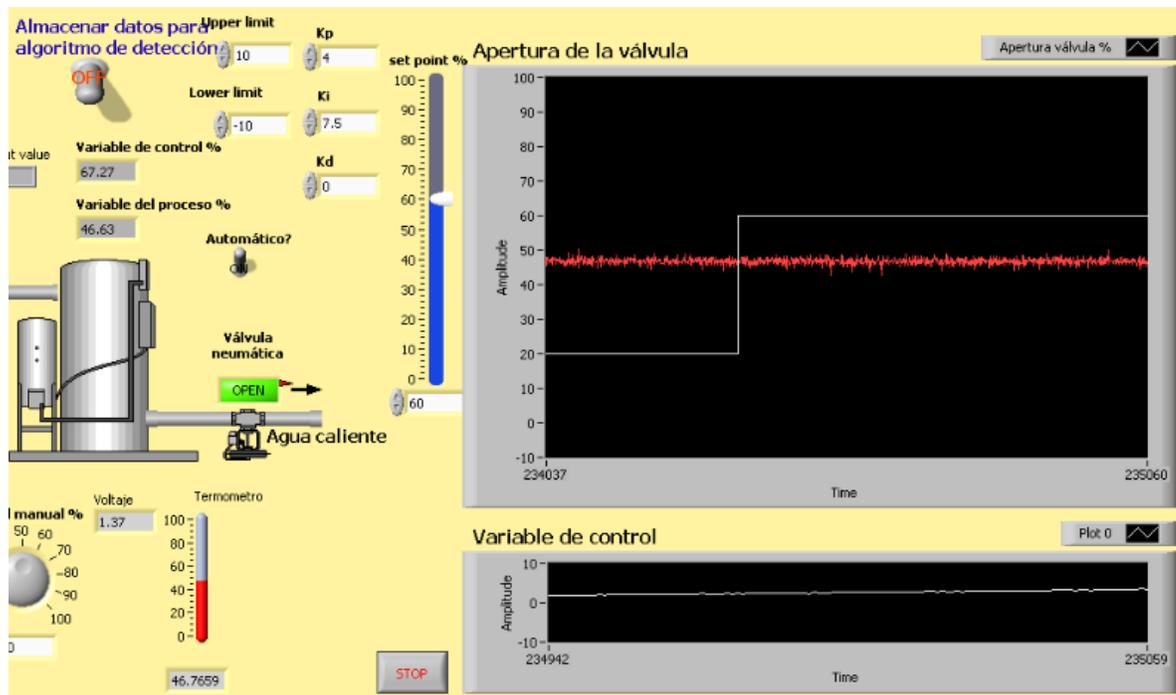


Fig. 4.20: Acción correctiva del sistema al variar el set-point

4.7 Análisis de resultados

En las pruebas se observó que el tiempo de establecimiento era aproximadamente de unos seis minutos. Esto es debido al comportamiento inherente que presentan los procesos de control de temperatura. Una de las razones de este comportamiento es debido a que la temperatura del líquido del proceso cerca a las paredes del intercambiador continúa su aumento o decrecimiento por un período de tiempo más. [3]

En conclusión, se puede indicar que a través de la apertura de la válvula se puede contar con un control eficiente de la temperatura.

Las pruebas de detección de fallos en la válvula arrojaron que la instalada en la planta no presenta falla alguna, razón por la cual para validar el algoritmo de detección se hicieron pruebas en la planta de flujo del laboratorio de control, en

la cual aplicado el algoritmo se determinó que la válvula de esa planta presentaba un fallo correspondiente a fricción estática.

La interfaz desarrollada en LabVIEW permite determinar a través de las pruebas realizadas, que efectivamente la interfaz hombre máquina cumple con los requerimientos de funcionalidad que deben tener todos los software HMI, tales como el monitoreo y supervisión de variables, reportar el estado de los componentes en caso este se encuentre bajo el efecto de una falla, acción de control sobre el proceso y registro histórico de las variables de la planta.



CONCLUSIONES

Se ha llevado a cabo el desarrollo de una interfaz-hombre máquina aplicada a una planta intercambiador de calor, con lo cual se ha demostrado que el entorno propuesto, debido a su capacidad visual, así como también por las librerías desarrolladas, permite la fácil adquisición de datos y muestra de resultados.

La interfaz Hombre-Máquina desarrollada nos permite ofrecer información en tiempo real, acerca de las fallas ocurridas en las bombas y la válvula de la planta estudiada.

La interfaz desarrollada ha sido diseñada lo más sencilla posible, de esta forma se tuvo una buena interacción simplificando las tareas que el usuario debe realizar, tales como supervisión de las variables, almacenamiento de datos, etc. Asimismo se han propuesto librerías específicas para la interacción con el sistema de detección de fallas.

Esta interfaz destinada al monitoreo y control de la planta, permite la interacción entre el usuario y la planta objeto de estudio. Mediante el control manual o automático es posible actuar sobre las variables del proceso.

RECOMENDACIONES

En el presente trabajo de tesis se desarrolla una interfaz hombre máquina consistente, sin embargo se recomienda ahondar en los métodos experimentales de obtención de modelos matemáticos para procesos, con el fin implementar un algoritmo de control más sofisticado, independientemente de la funcionalidad y estética de la interfaz, debido al comportamiento complejo del sistema, como es el caso del control predictivo, basado en modelos que son capaces de contemplar las perturbaciones inherentes al sistema.

También sería interesante combinar los estudios de control y automatización con entornos de programación que permitan desarrollar interfaces de usuario orientados a aplicaciones industriales, así además de consolidar los conocimientos teóricos, se proporciona al alumno la herramienta con la cual se permitiría a un posible cliente ajeno al campo de control, la interacción con una planta específica.

BIBLIOGRAFÍA

[1] J. J. Gertler. Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, 1998.

[2] Ramírez Cavassa, César. Ergonomía y productividad. Limusa, 2008.

[3] Ing. Carranza Noriega, Raymundo. Automatización: Tópicos de Instrumentación y Control. Pontificia Universidad Católica del Perú, 2001.

[4] L. Arenas, A. Castilla y D. Rojas

Interfaz hombre-máquina. Colombia: Universidad Autónoma del Caribe, Facultad de Ingeniería. Consulta: 10 de Mayo de 2012.

<http://es.scribd.com/doc/85749234/Interfaz-Hombre-Maquina-HMI>

[5] iaci.unq.edu.ar

Introducción a HMI (Interfaz Hombre-Máquina). Argentina: Universidad de Quilmes. Consulta 12 de Mayo de 2012.

<http://iaci.unq.edu.ar/materias/laboratorio2/HMI%5CIntroduccion%20HMI.pdf>

[6] Marcos Mora, Mari Carmen

Presente, pasado y futuro de la investigación en interacción hombre-ordenador aplicada a la documentación. España: Universidad de Zaragoza, Facultad de Ciencias. Consulta 15 de Mayo de 2012.

<http://es.scribd.com/doc/39106686/Interaccion-Hombre-Ordenador>

[7] <http://www.lsi.us.es>

Diseño de Interfaces Hombre-Máquina. España: Universidad de Sevilla, Departamento de Lenguajes y Sistemas Informáticos. Consulta 15 de Mayo de 2012.

http://www.lsi.us.es/docencia/pagina_asignatura_doc.php?id=23&cur=2007

[8] Arellano, Gabriel E.

Interfaz hombre-máquina. Uruguay: Universidad tecnológica regional, Facultad Regional Concepción del Uruguay. Consulta: 10 de Mayo de 2012.

<http://www.slideshare.net/areche/interfaz-hobremquina-presentation>

[9] Medina, Santiago.

Diseño de la interfaz de usuario. Ecuador: Universidad técnica particular de Loja, Facultad de Ingeniería de Software. Consulta: 23 de Mayo de 2012.

<http://www.slideshare.net/sanfer08/disen-de-la-interfaz-de-usuario>

[10] Rodeiro Iglesias, Javier

Diseño de Interfaces Hombre-Máquina. España: Universidad de Vigo. Consulta: 22 de Mayo de 2012.

<http://trevinca.ei.uvigo.es/~jrodeiro/Teaching/diu/DIU-Tema1.pdf>

[11] C. Roberto, P. Alex. Computer Vision for Human-Machine Interaction. Cambridge University Press, 1998.

[12] R. Galeas Arana. Programación gráfica con LabVIEW. Grupo editorial Megabyte, 2011.

[13] Lajara. J, Pelegrí. J. LabVIEW: Entorno Gráfico de Programación 2a edición. Marcombo, 2011.

[14] <http://home.hit.no/~hansha/>

Data Acquisition in LabVIEW. Noruega: Telemark University College, Department of Electrical Engineering, Information Technology and Cybernetics. Consulta: 11 de Junio de 2012.

<http://home.hit.no/~hansha/documents/labview/training/Data%20Acquisition%20in%20LabVIEW/Data%20Acquisition%20in%20LabVIEW.pdf>

[15] Velázquez. I

Acondicionamiento de señales. Chile: Universidad de la Frontera, Facultad de Ingeniería Eléctrica. Consulta 12 de Mayo de 2012.

http://www.inele.ufro.cl/apuntes/Instrumentacion_y_Control/Ivan_Velazquez/Catdra/Capitulo%203.%20Acondicionamiento%20de%20senales.pdf

[16] Bishop, Robert H. LabVIEW 8 Student Edition. Pearson Prentice Hall, 2007.

[17] Gómez. M, Zabala. G, Dávila. J

Uso de LabVIEW para sistemas de control en Ingeniería Química. Colombia: Universidad de los Andes, Facultad de Ingeniería Química. Consulta 14 de Setiembre de 2012.

<http://www.scielo.org.co/pdf/dyna/v78n169/a17v78n169.pdf>

[18] Andrade. R, Macas. W. Diseño y Construcción de un módulo didáctico para medir y controlar caudal y nivel. Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica. Consulta: 20 de Setiembre de 2012.

<http://es.scribd.com/doc/57161316/57/Control-Proporcional-Integral-Derivativo-PID-10>

[19] Oncins, M

Mandos y señales: ergonomía de percepción. España: Centro Nacional de Condiciones de Trabajo.

http://www.jmcprl.net/ntps/@datos/ntp_241.htm